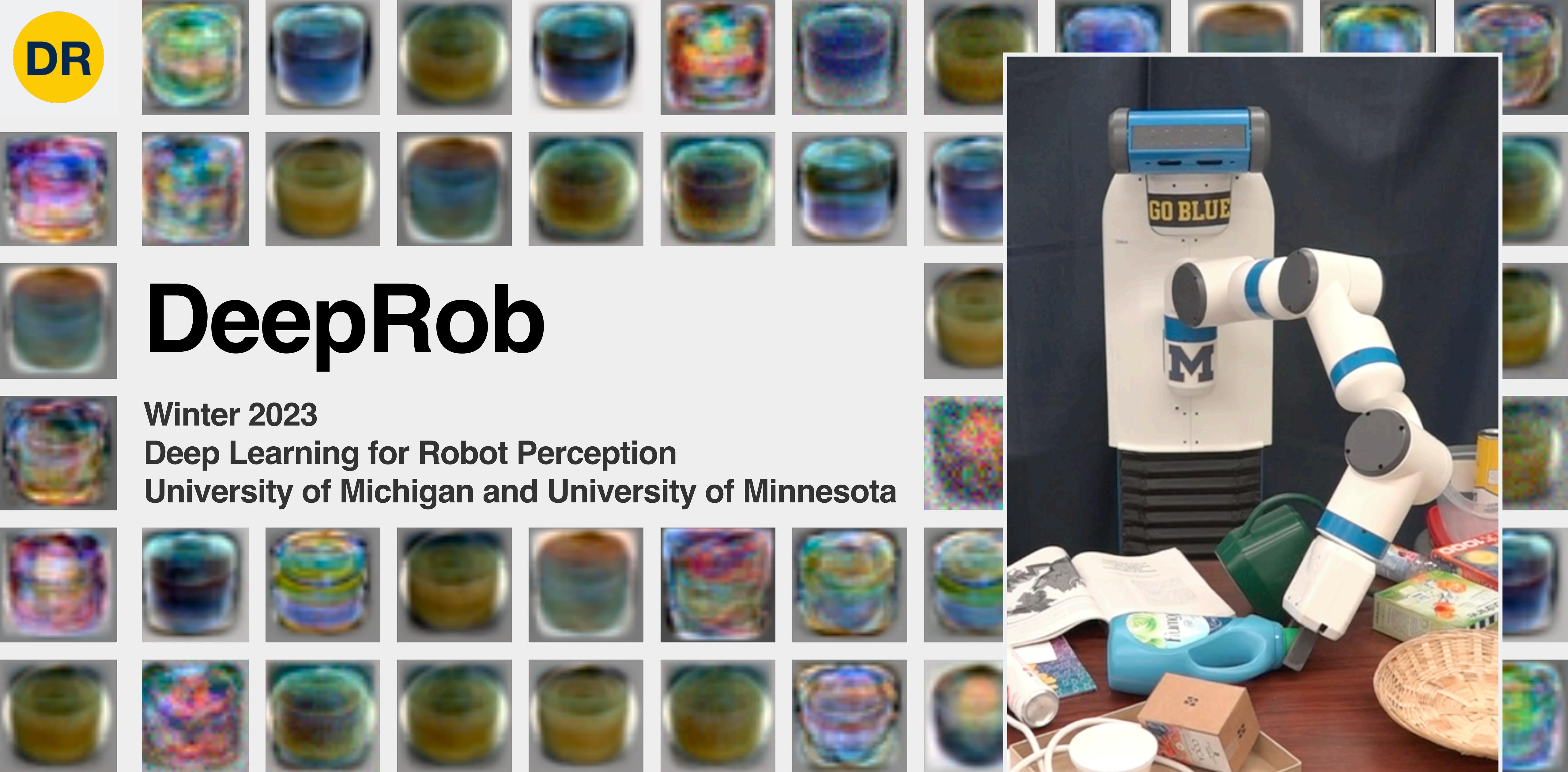


DR



DeepRob

Winter 2023

Deep Learning for Robot Perception

University of Michigan and University of Minnesota



DR

4Progress



DR

4Progress

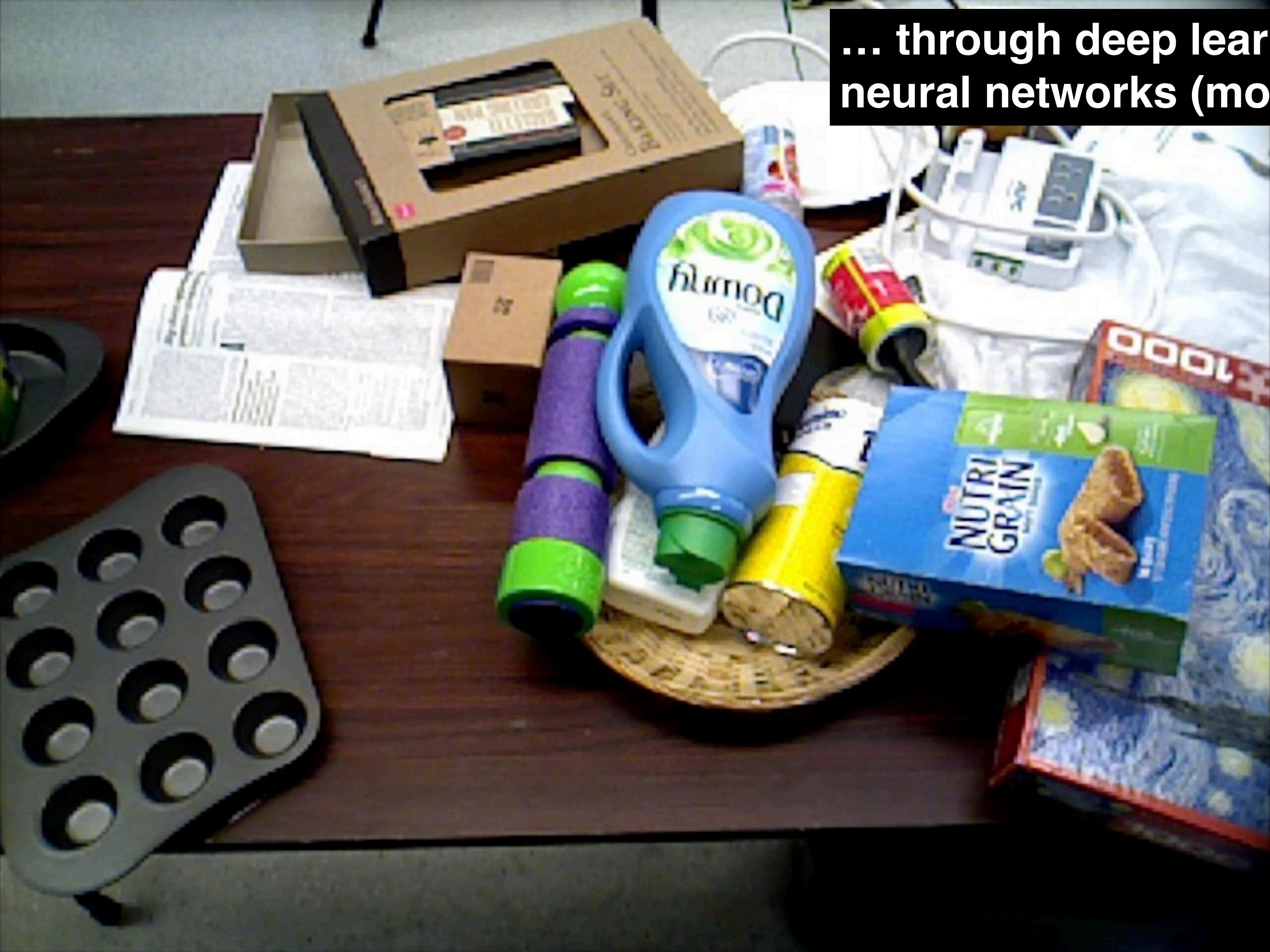
Sui, Zhou, Zeng, Jenkins 2017

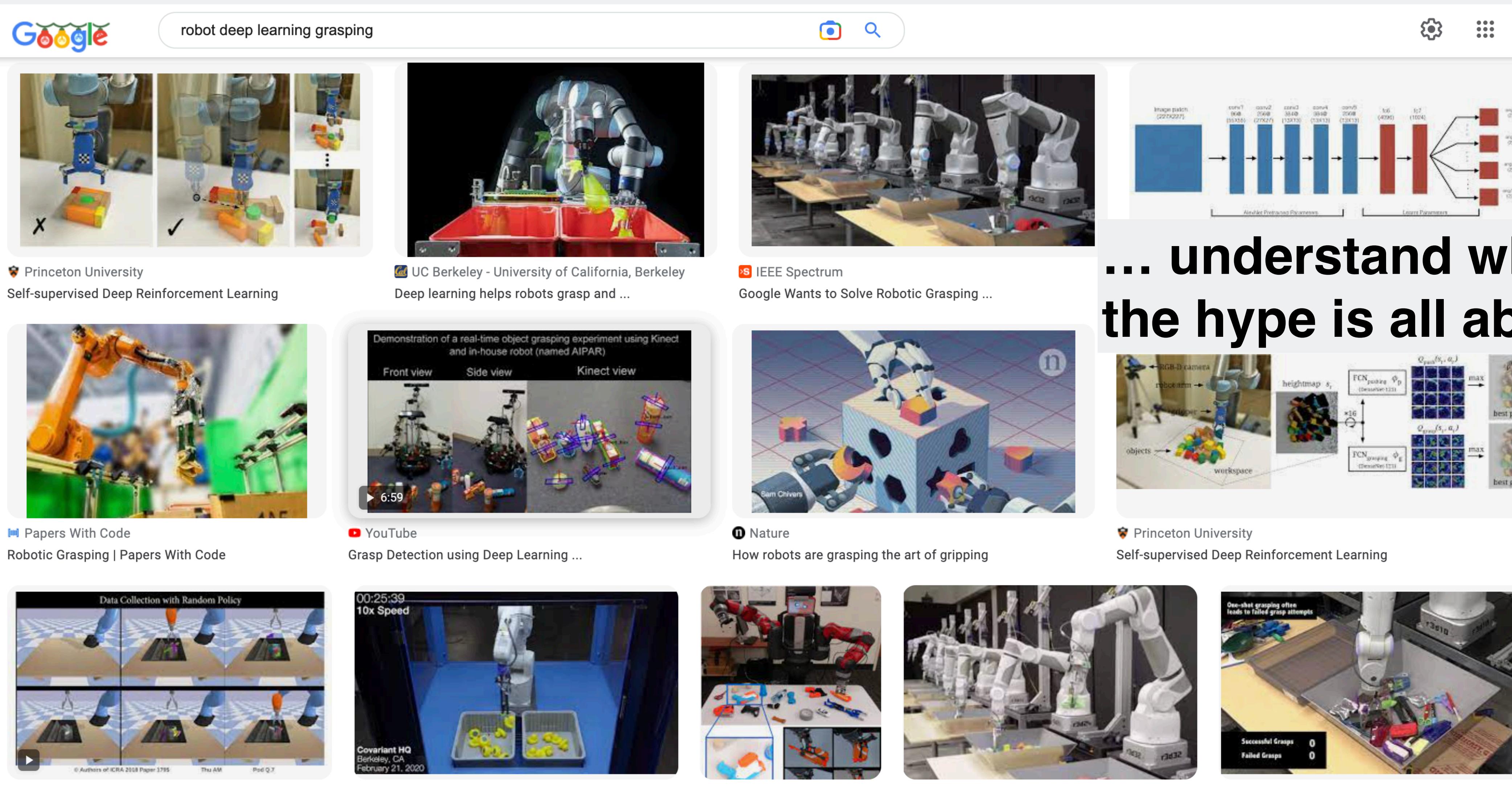
Our goal:
Perception and manipulation
of objects in clutter...



DR

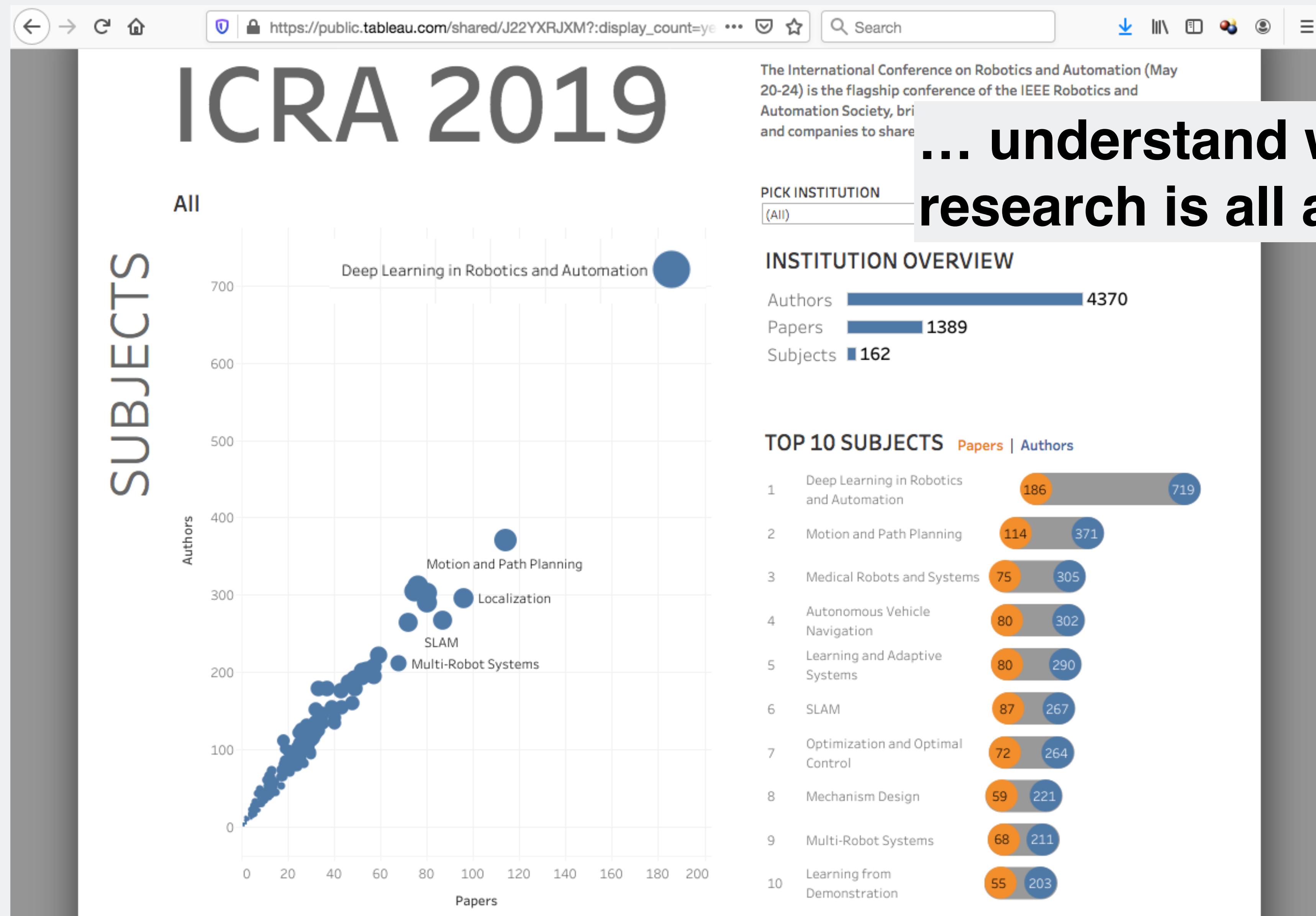
... through deep learning with
neural networks (mostly) ...





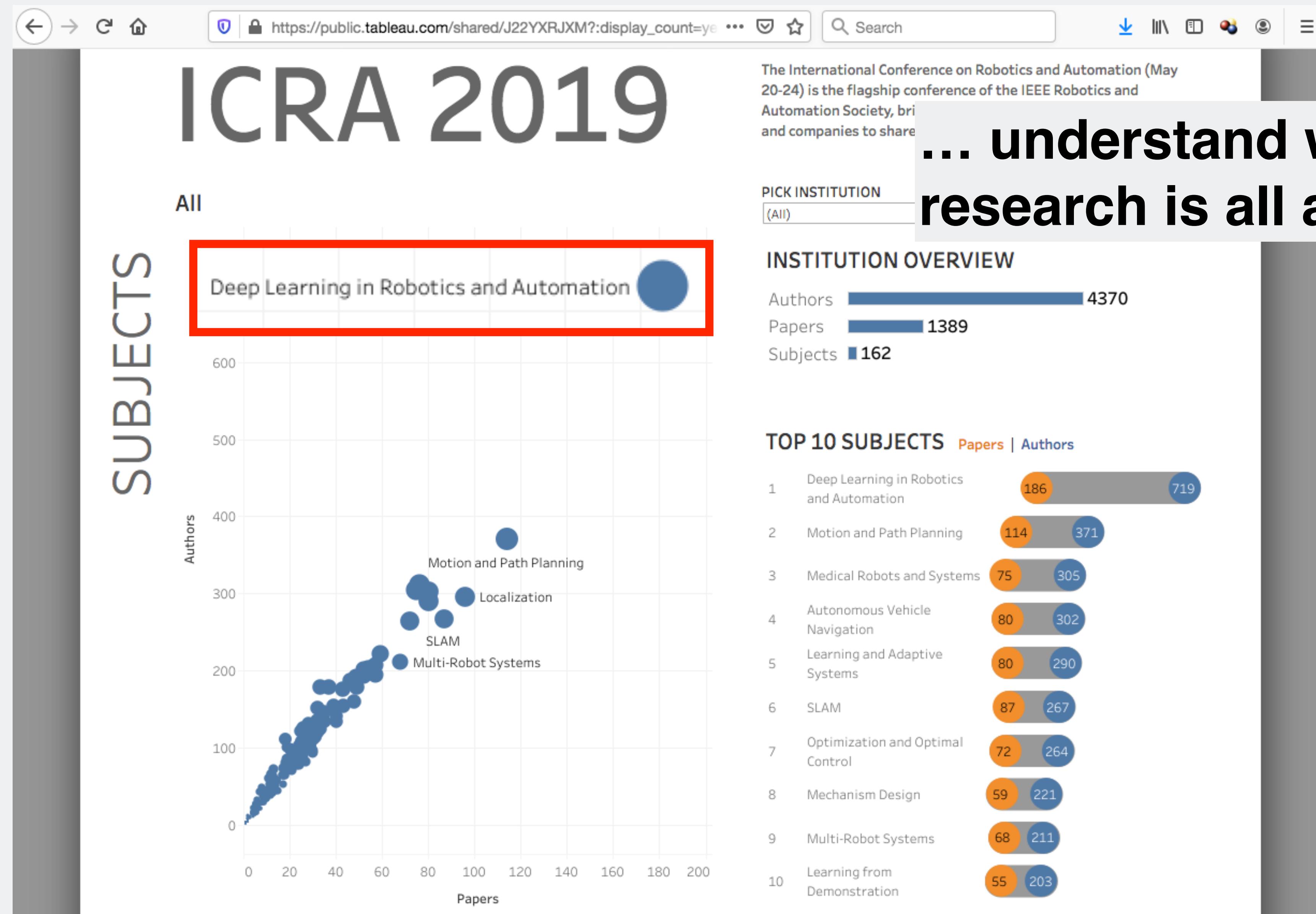
... understand what
the hype is all about ...





... understand what the research is all about ...



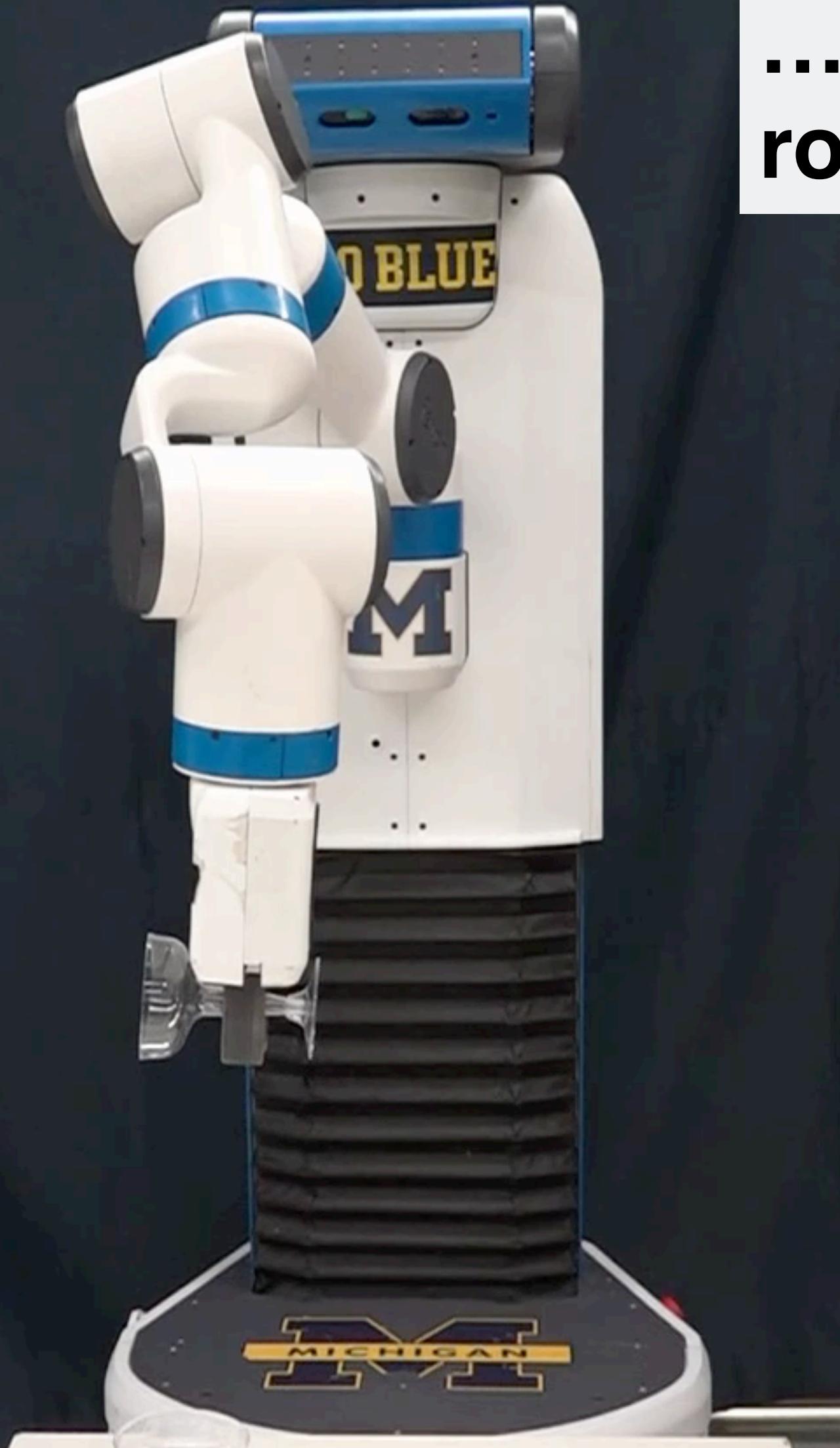


... understand what the research is all about ...



DR
8x

... and generate new ideas for robotics research (hopefully)



LIT

[Zhou, Chen, Jenkins *IEEE RA-L 2020*]



Welcome to DeepRob!



DR

Welcome to DeepRob!



Anthony Opiari



Prof. Chad Jenkins



Prof. Karthik Desingh



DeepRob Course Staff



Anthony Opiari



Prof. Chad Jenkins



Prof. Karthik Desingh



Huijie Zhang
Research Associate



Jiyue Zhu
Instructional Aide



DeepRob is a fork (and a star) of Deep Learning for Computer Vision

EECS 498.008 / 598.008
Deep Learning for Computer Vision
Winter 2022

Schedule

Lectures are Mondays and Wednesdays, 4:30pm to 6pm. Attendance is not required. Recordings will be posted after each lecture in case you are unable to attend the scheduled time.

We will hold lectures remotely over Zoom for the first few weeks. See Canvas for the Zoom meeting information.

Some lectures have reading drawn from the course notes of [Stanford CS 231n](#), written by [Andrei Karpathy](#).

Some lectures have optional reading from the book *Deep Learning* by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (GBC for short). The entire book is [available for free online](#) so you don't need to buy a copy.

Event	Date	Description	Course Materials
Lecture 1 [Remote]	Wednesday January 5	Course Introduction Computer vision overview Historical context Course logistics	[slides] [video (UMich only)] [FA2019 video (public)] [Python tutorial] [GBC Sec 1.2] [GBC Sec 6.6]
Lecture 2 [Remote]	Monday January 10	Image Classification Data-driven approach K-Nearest Neighbor Hyperparameters Cross-validation	[slides] [video (UMich only)] [FA2019 video (public)] [231n Image Classification]
Lecture 3 [Remote]	Wednesday January 12	Linear Classifiers Algebraic / Visual / Geometric viewpoints Softmax / SVM classifiers	[slides] [video (UMich only)] [FA2019 video (public)] [231n Linear Classification]

Assignment 1 Due

**Similar introductory projects,
New projects for 3D estimation,
Build up to mini research project**

CS231n Home **Course Notes** **Coursework** **Schedule** **Office Hours** **Final Projects** **Ed**

CS231n: Deep Learning for Computer Vision
Stanford - Spring 2022



*This network is running live in your browser

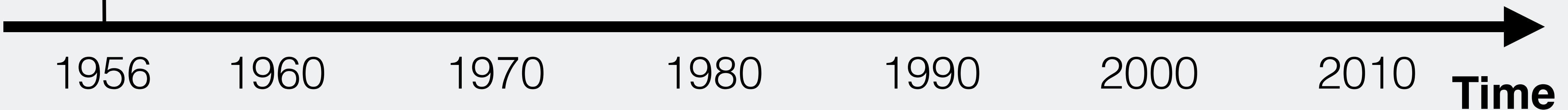
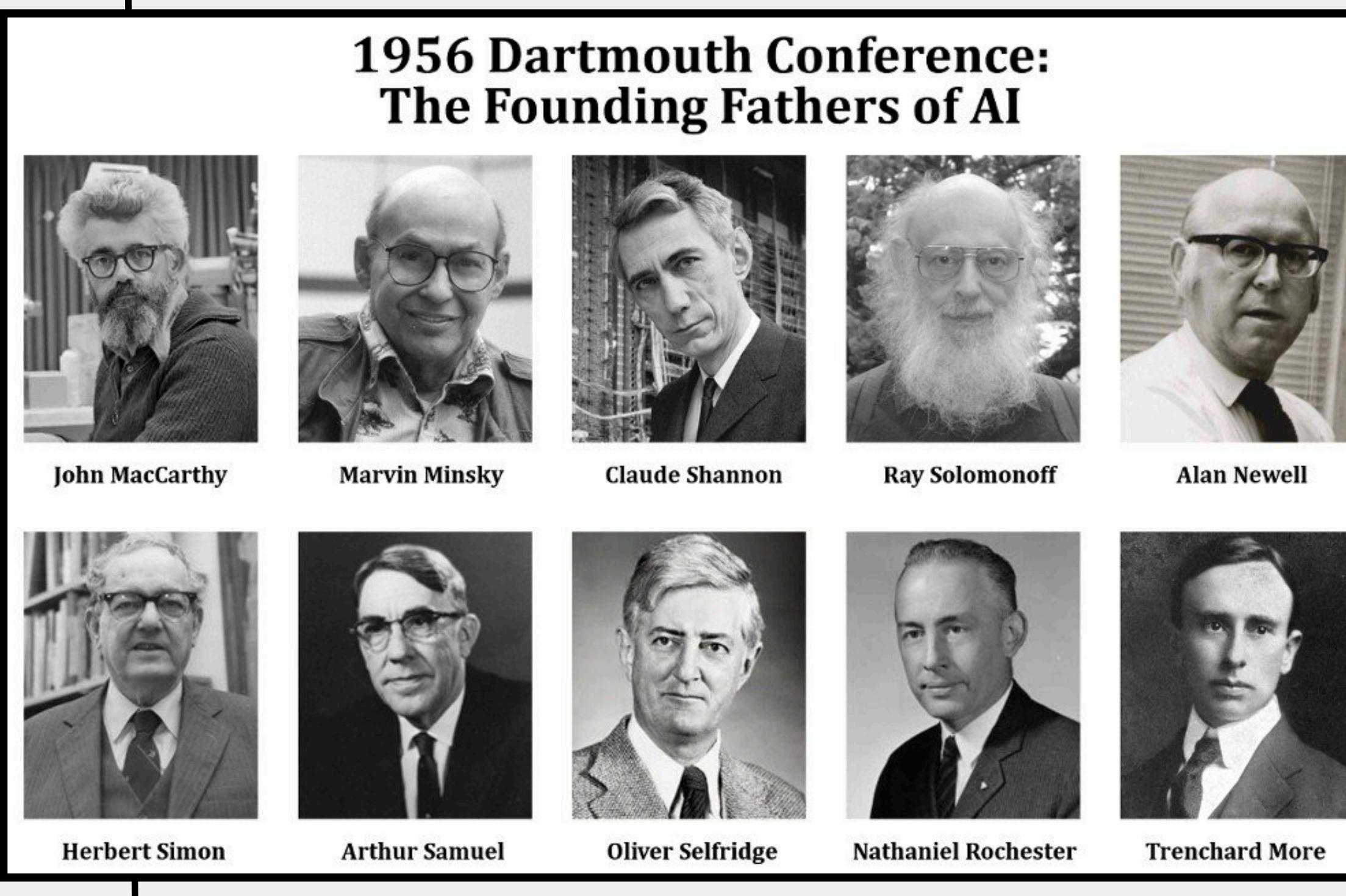
Course Description

Computer Vision has become ubiquitous in our society, with applications in search, image understanding, apps, mapping, medicine, drones, and self-driving cars. Core to many of these applications are visual recognition tasks such as image classification, localization and detection. Recent developments in neural network (aka "deep learning") approaches have greatly advanced the performance of these state-of-the-art visual recognition systems. This course is a deep dive into the details of deep learning architectures with a focus on learning end-to-end models for these tasks, particularly image classification. During the 10-week course, students will learn to implement and train their own neural networks and gain a detailed understanding of cutting-edge research in computer vision. Additionally, the final assignment will give them the opportunity to train and apply multi-million parameter networks on real-world vision problems of their choice. Through multiple hands-on assignments and the final course project, students will acquire the toolset for setting up deep learning tasks and practical engineering tricks for training and fine-tuning deep neural networks.

[Previous Offerings](#)

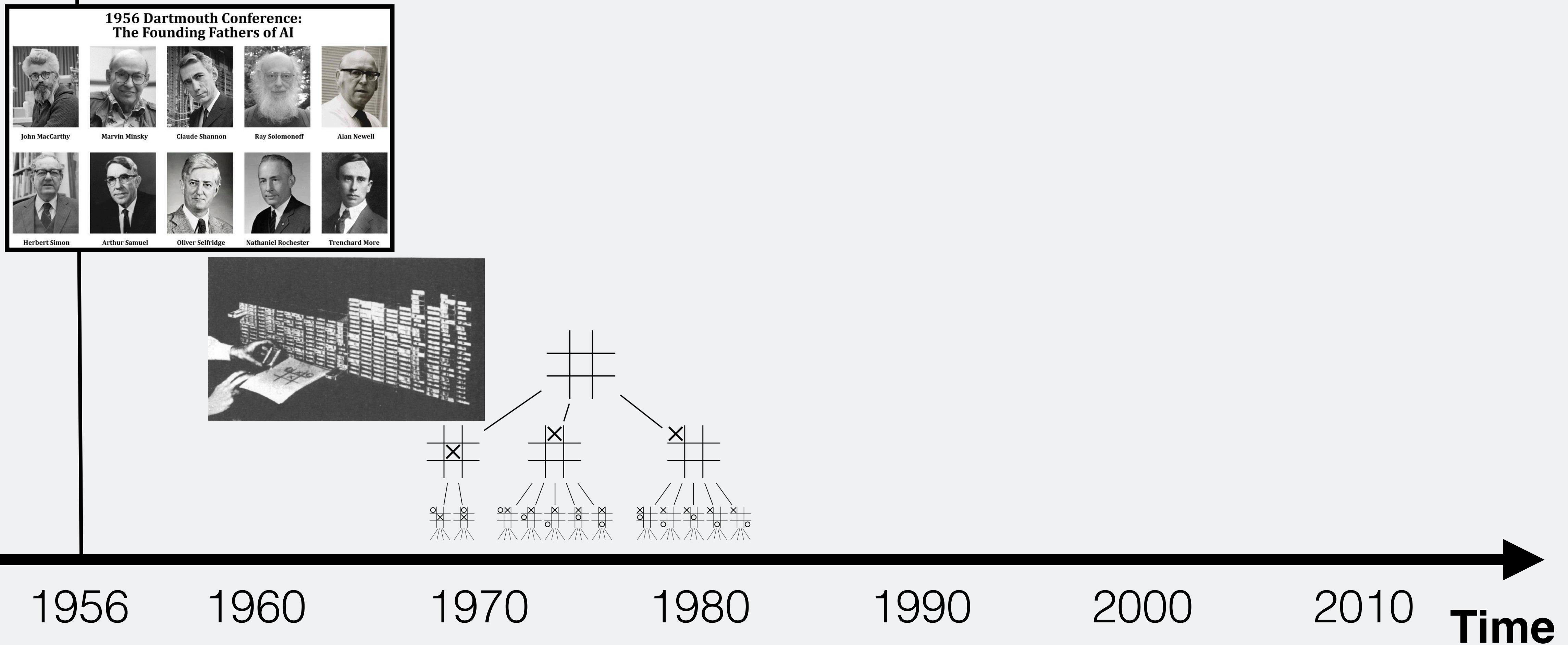


A brief history of AI



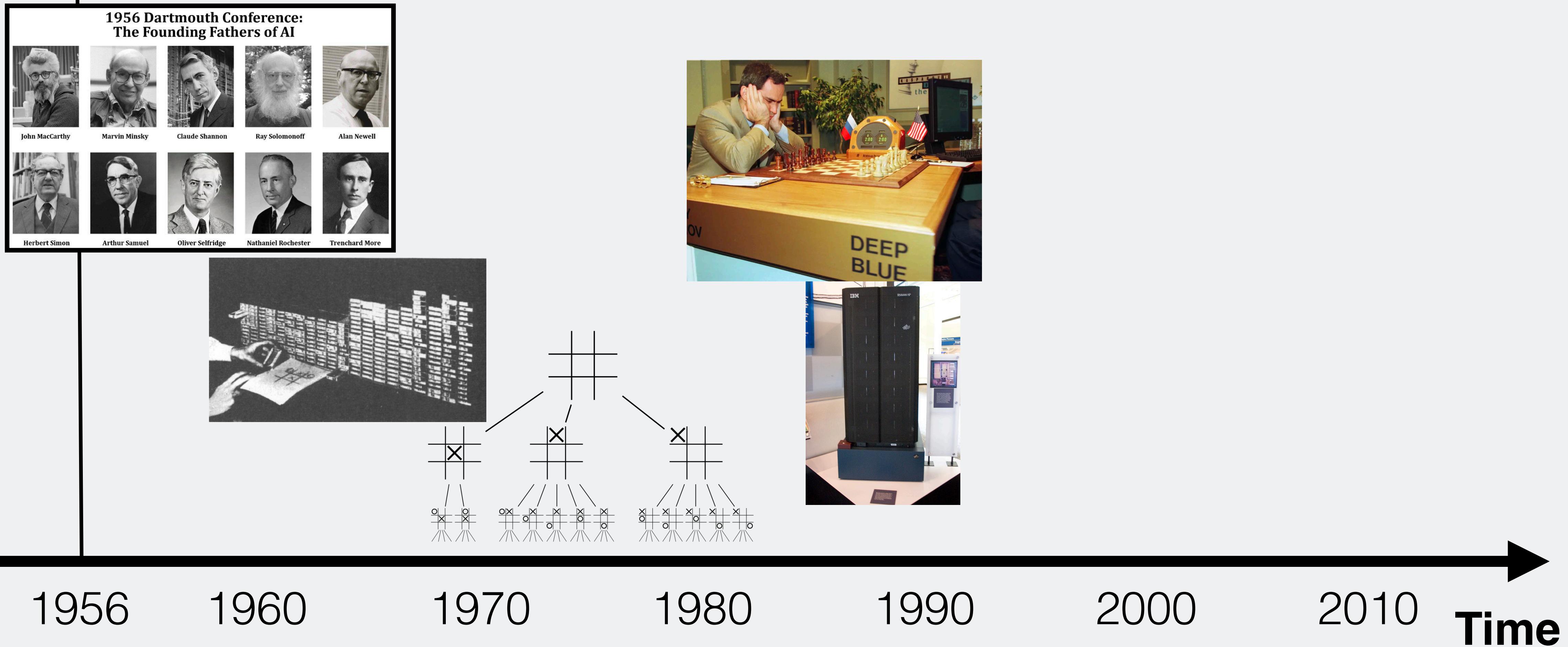
First wave AI: Model-based

“Think through the entire problem”



First wave AI: Model-based

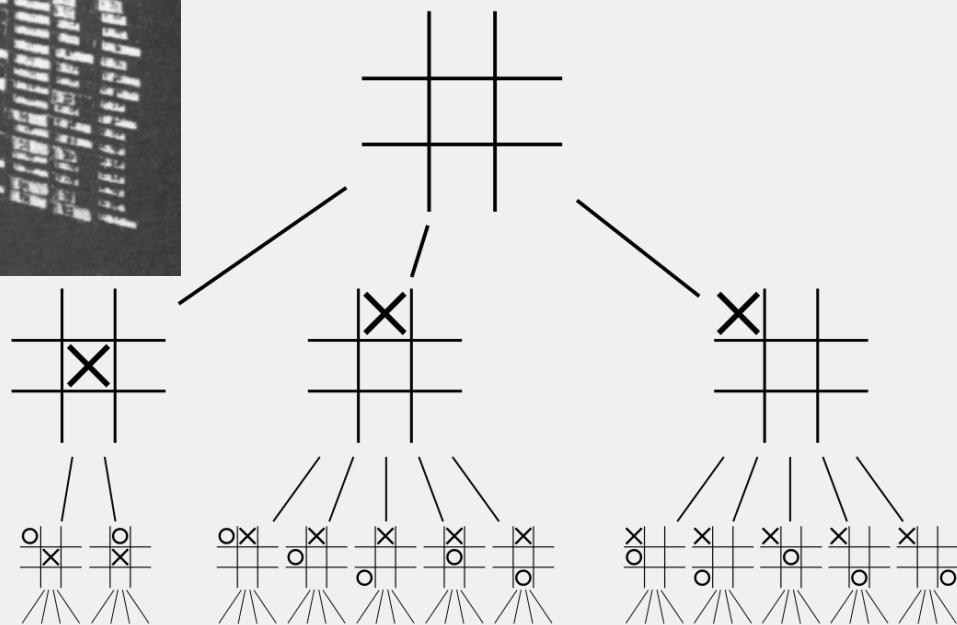
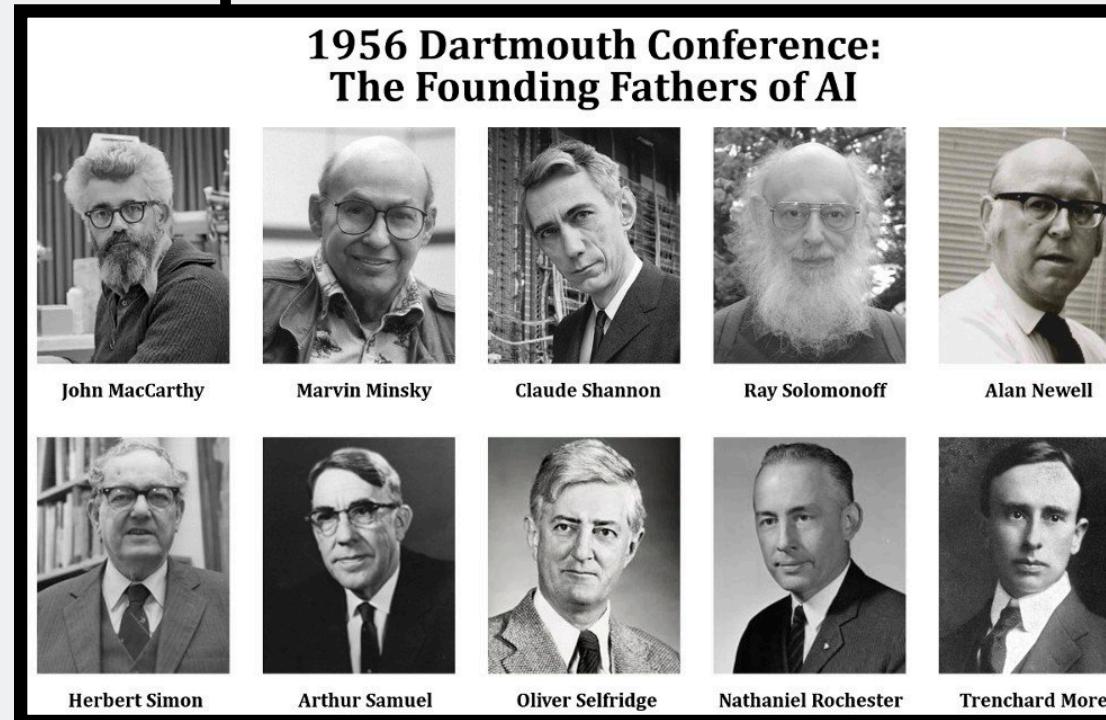
“Think through the entire problem”





First wave AI: Model-based

“Think through the entire problem”



1956

1960

1970

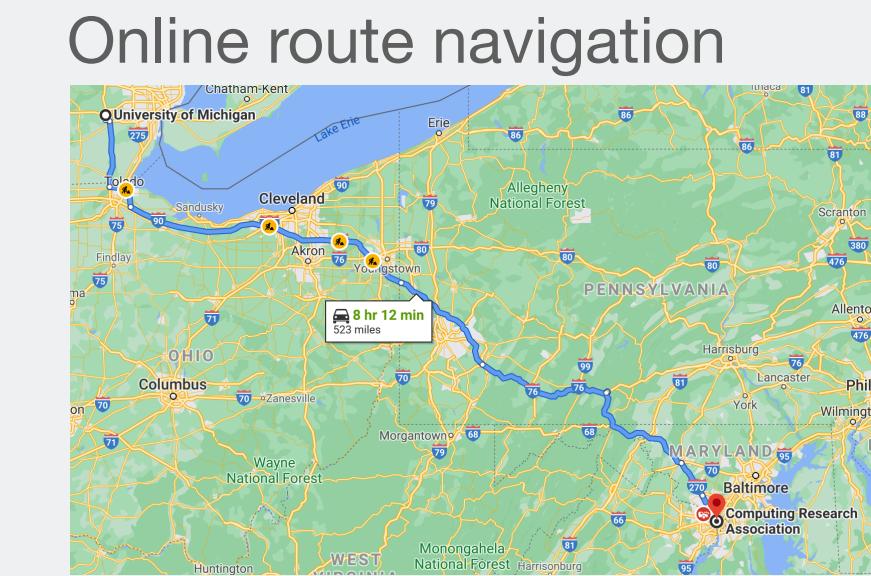
1980

1990

2000

2010

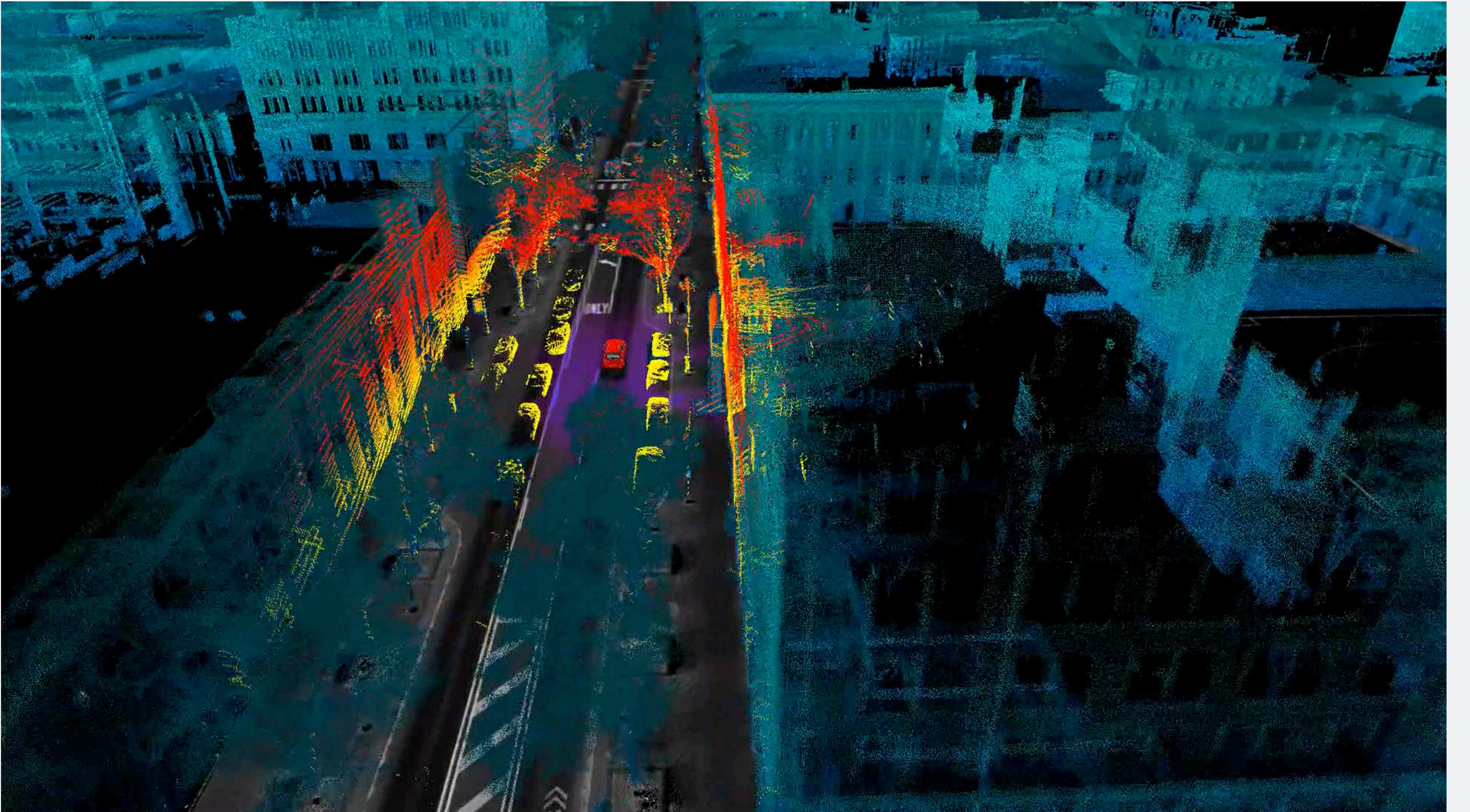
Time



Self-driving cars



DR

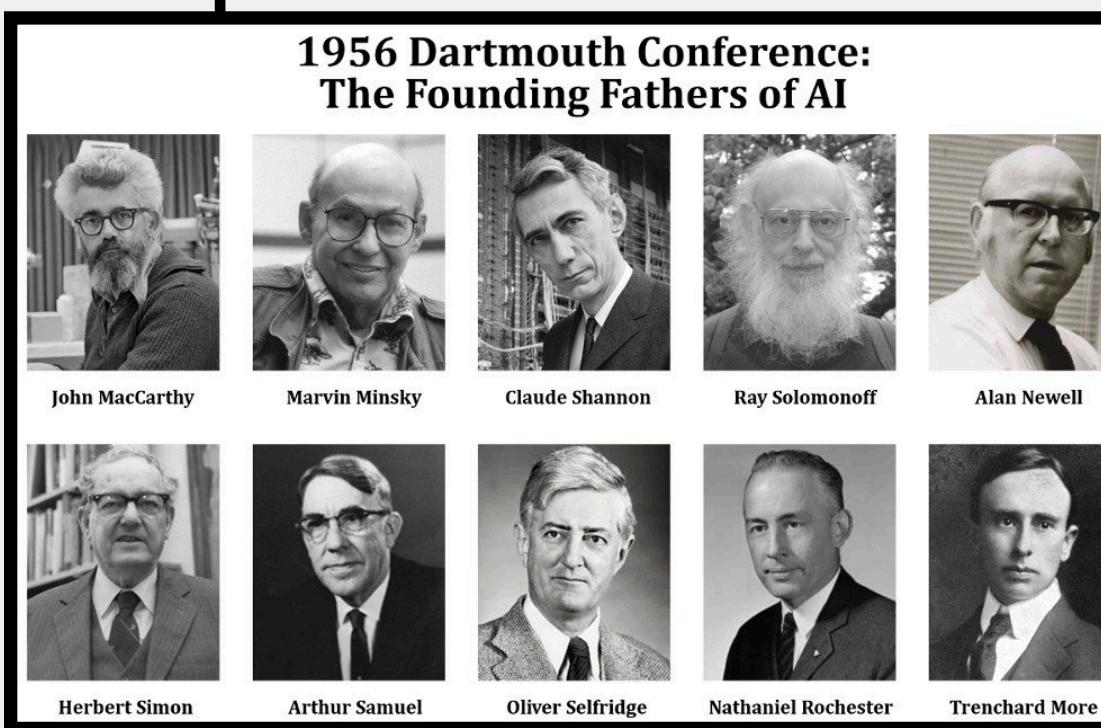


Let's watch this video closely



First wave AI: Model-based

“Think through the entire problem”

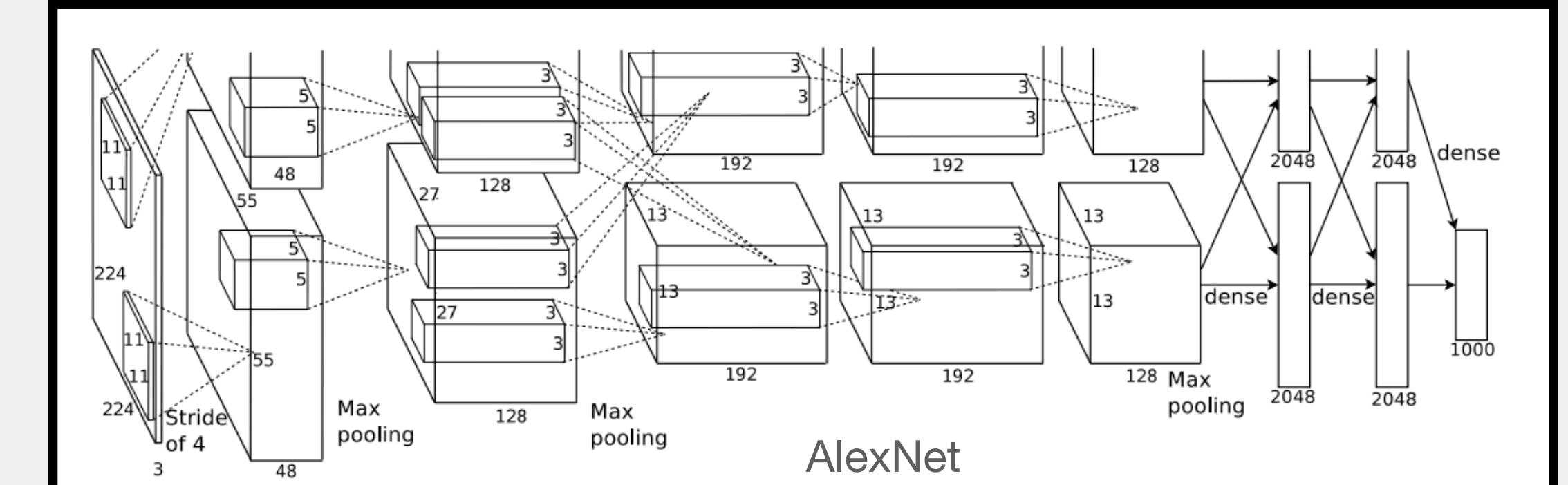


1956

Second wave AI: Data-driven

“Learn from lots of data”

Rise of “deep learning”



2011

Time



First wave AI: Model-based

The screenshot shows a news article from IEEE Spectrum. The title is "Deep Learning Makes Driverless Cars Better at Spotting Pedestrians". Below the title is a subtitle: "Pedestrian detection systems for cars could become faster and more accurate with help from deep learning algorithms". The author is listed as "By Jeremy Hsu". Below the text are four small images showing pedestrian detection results on city streets. The bottom caption reads "Images: Statistical Visual Computing Lab/UC San Diego". The navigation bar at the top includes links for Engineering Topics, Special Reports, Blogs, Multimedia, The Magazine, Professional Resources, and Search.

1956



Second wave AI: Data-driven

"Learn from lots of data"



"deep learning"

2011

Time



TEXT PROMPT
an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

Edit prompt or view more images↓

TEXT PROMPT
an armchair in the shape of an avocado....

AI-GENERATED IMAGES

Edit prompt or view more images↓

DALL-E (OpenAI)

Second wave AI: Data-driven

“Learn from lots of data”



“deep learning”

2011

Time





TEXT PROMPT
an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

Your AI pair programmer

GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time, right from your editor.

[Start my free trial >](#) [Explore docs](#)

TEXT PROMPT
an armchair in the shape of a pear

AI-GENERATED IMAGES

```
ts sentiments.ts  oo write_sql.go  + parse_expenses.py  addresses.rb
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch(`http://text-processing.com/api/sentiment/`, {
9     method: "POST",
10    body: `text=${text}`,
11    headers: {
12      "Content-Type": "application/x-www-form-urlencoded",
13    },
14  });
15  const json = await response.json();
16  return json.label === "pos";
17 }
```

DALL-E (OpenAI)

Copilot (OpenAI)

Second wave AI: Data-driven

“Learn from lots of data”



“deep learning”

2011

Time





Second wave AI: Data-driven

TEXT PROMPT
an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

Your AI pair programmer

GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in

In the following sample, ChatGPT is able to understand the reference ("it") to the subject of the previous question ("fermat's little theorem").

Sample 3 of 4 Previous Next

User what is the fermat's little theorem

ChatGPT Fermat's Little Theorem states that if p is a prime number and a is an integer not divisible by p , then a^p (a to the power p) is congruent to a modulo p . In other words, if we divide a^p by p , the remainder is always a .

For example, if $p = 5$ and $a = 2$, then $2^5 = 32$, and 32 divided by 5 has a remainder of 2. Therefore, 2^5 is congruent to 2 modulo 5, and Fermat's Little Theorem holds for this case.

Fermat's Little Theorem is often used in cryptography and other applications where it is necessary to perform modular arithmetic operations quickly and efficiently. It is also a useful tool for proving other theorems in number theory

User how is it used in cryptography?

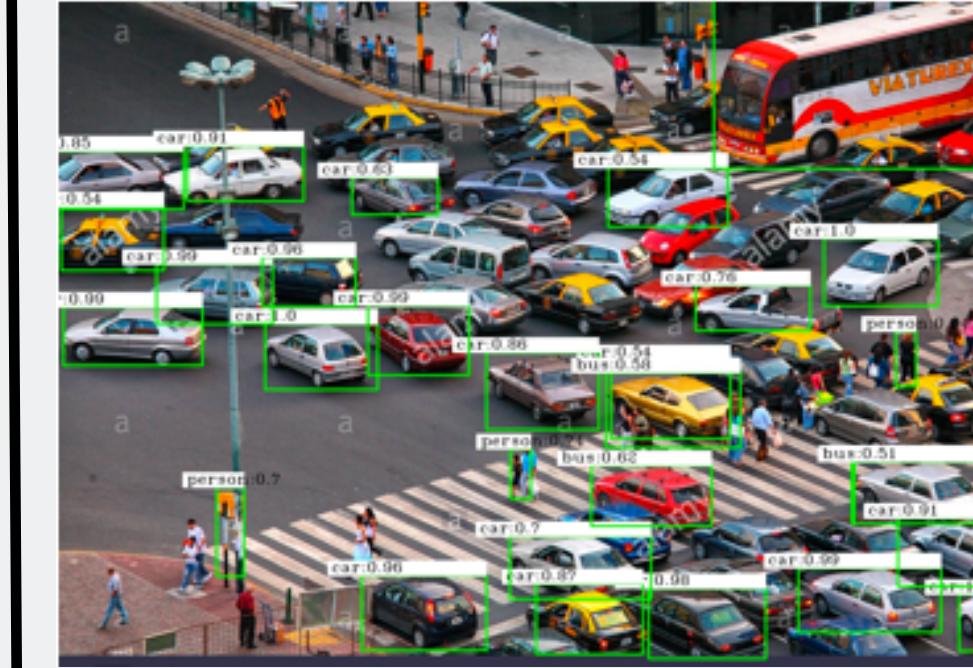
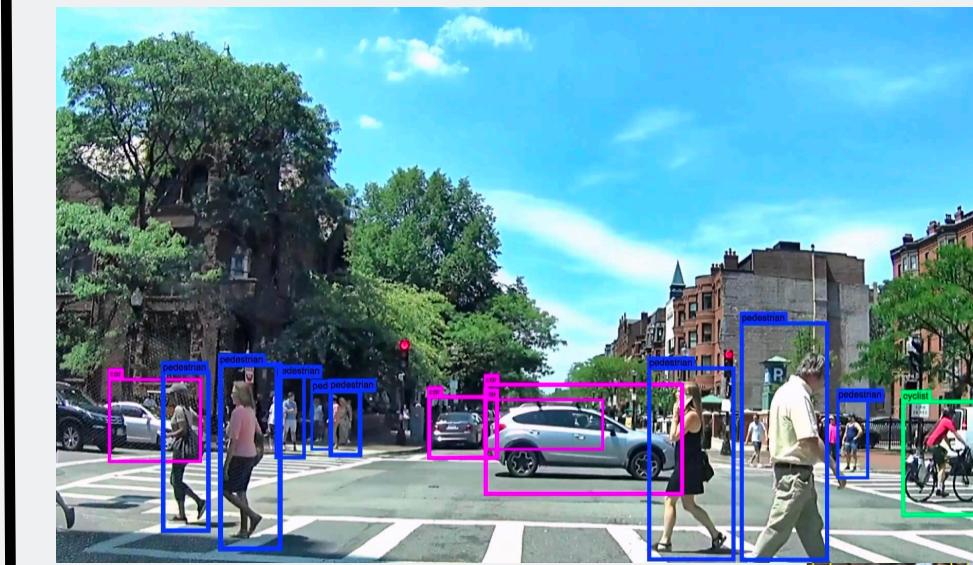
ChatGPT Fermat's Little Theorem is used in cryptography in several ways. One of the most common applications is in the generation of so-called "public-key" cryptography systems, which are used to securely transmit messages over the internet and other networks.

In a public-key cryptography system, each user has a pair of keys: a public key, which is widely known and can be used by anyone to encrypt a message intended for that user, and a private key, which is known only to the user and is used to decrypt messages that have been encrypted with the corresponding public key.

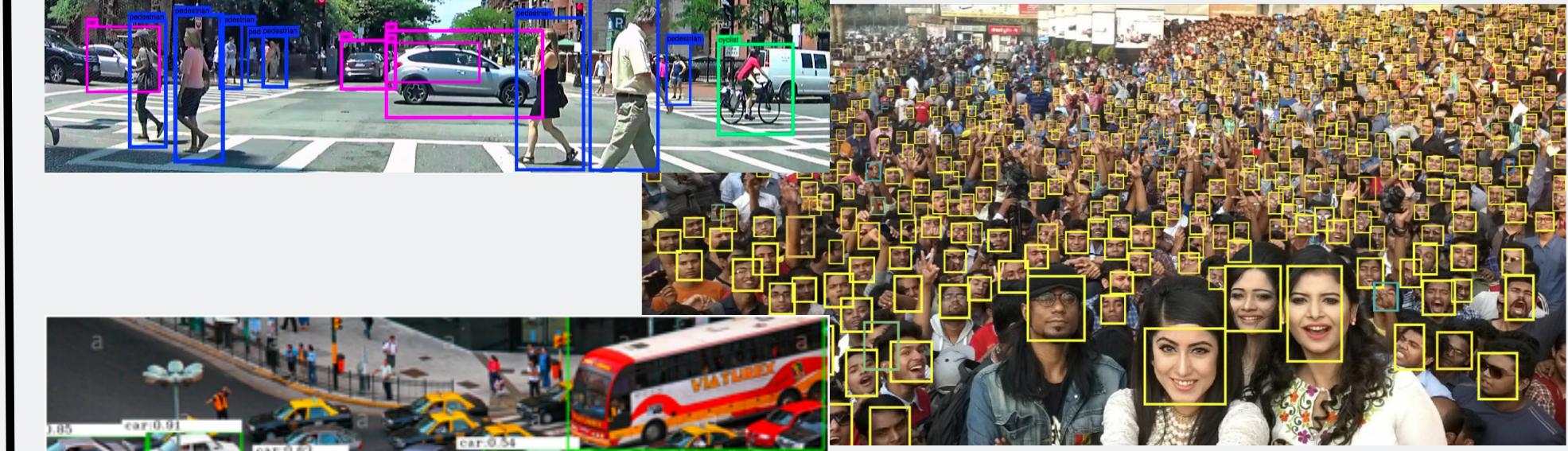
DALL-E (OpenAI)

Copilot (OpenAI)

ChatGPT (OpenAI)



"Learn from lots of data"



"deep learning"

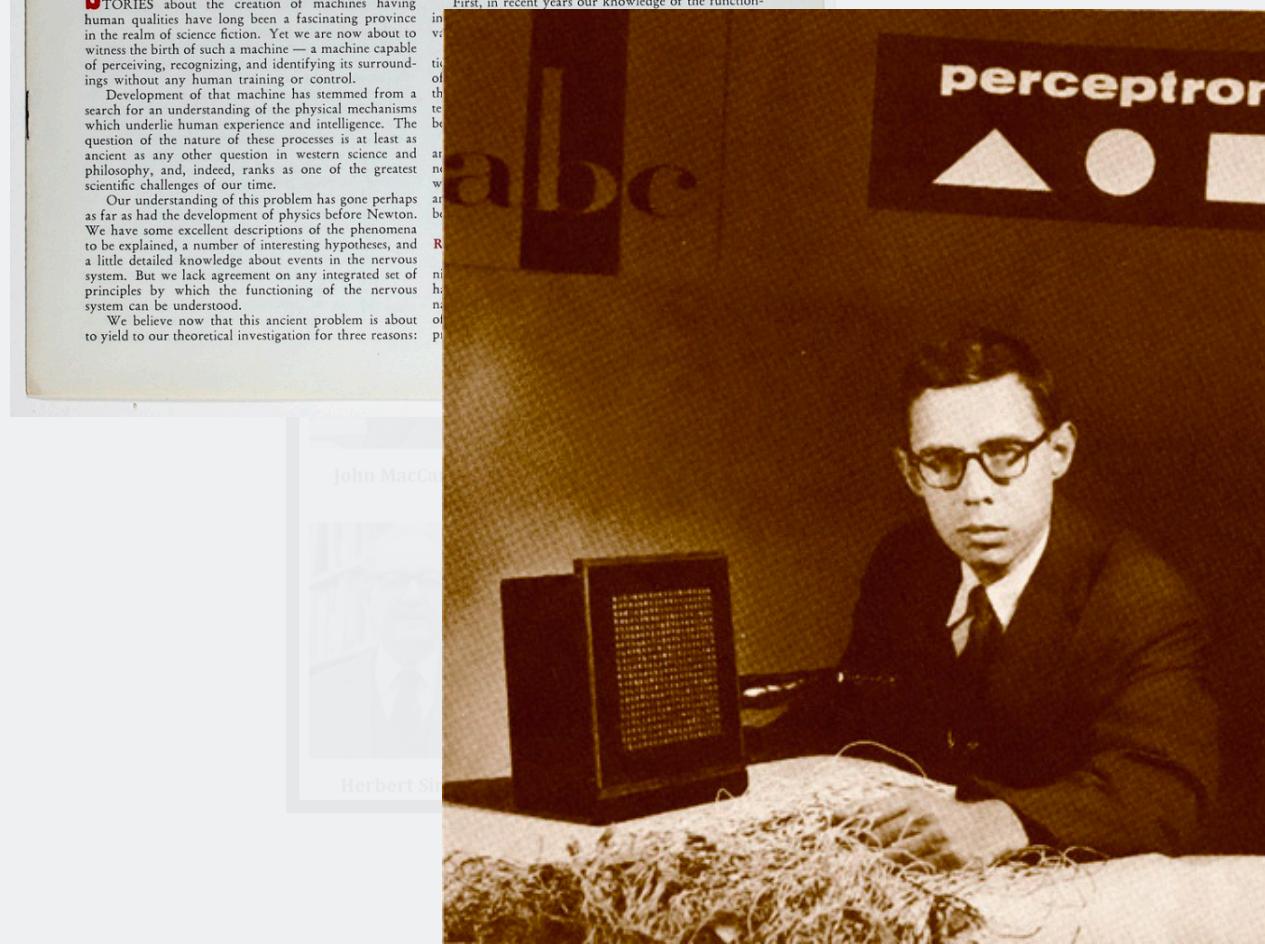
2011

Time





The AI of today needed decades of investment and research



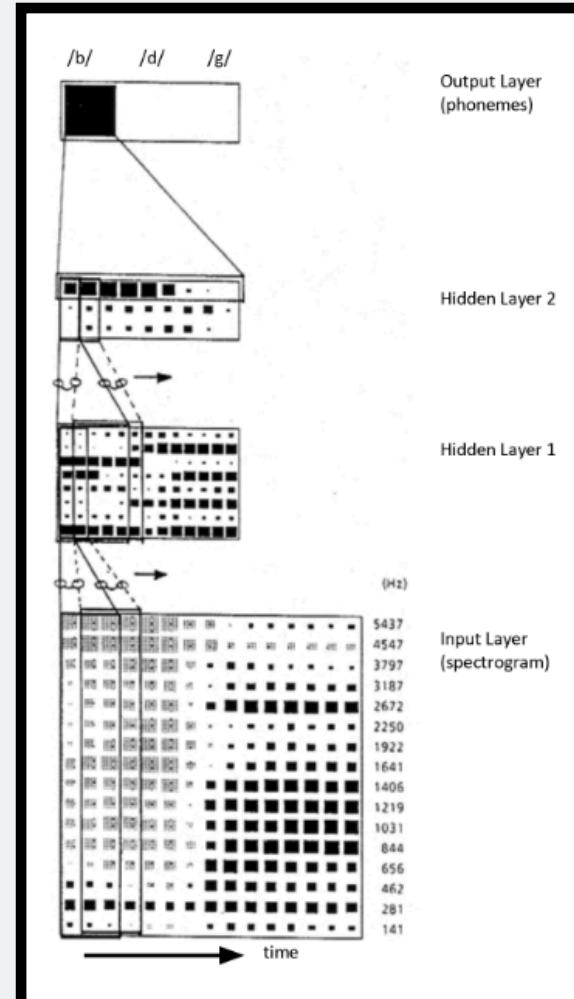
Rosenblatt's Perceptron

1956

1958



Speech recognition by neural networks



Waibel et al.

1989

2011

Second wave AI: Data-driven

"Learn from lots of data"



"deep learning"



Time





What is this?

Search

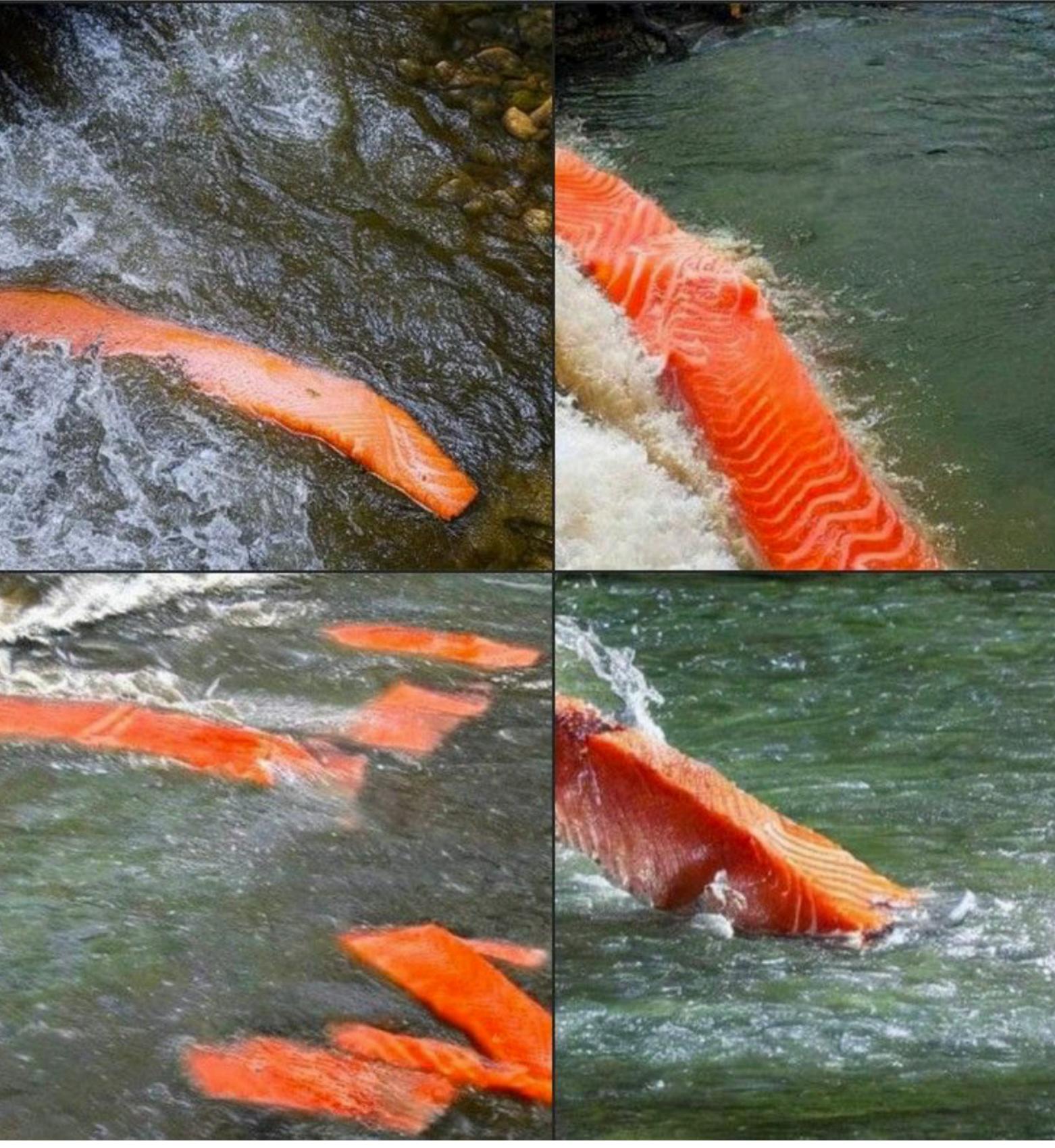
Carlos la Orden Tovar · 3rd+ Trainer and speaker at international events in the language industry 2d

+ Follow

They asked an AI engine to recreate "a salmon swimming down a river", and here's what it guessed it would look like.

It's exactly the same thing that happens to your texts when you leave them in the hands of a raw machine translation engine and fail to plan a proper post-editing, proofreading and—above all—put a discerning human eye in the loop to look for context and read between the lines.

#IA #MT #MachineTranslation



Be mindful about how you use AI

“a salmon swimming down a river”





Posted by u/Callec254 25 days ago

4 Fluff

I asked the new Open AI GPT chat bot for advice...

How do you counter mass Protoss air as a Terran player in Starcraft II?

As a Terran player facing a mass Protoss air army in Starcraft II, one potential counter strategy is to use missile turrets. These turrets can be built on the Terran player's own base, and are effective at shooting down enemy air units.

In addition to missile turrets, the Terran player can also use ground units such as marines, marauders, and thors to attack the Protoss air units. These units are effective at dealing damage to air units, and can be used to weaken the enemy army before it reaches the Terran player's base.

Another potential counter strategy is to use Vikings, which are Terran air units that are specifically designed to counter other air units. These units can be used to attack the Protoss air army directly, or to provide support for other ground units.

Overall, the key to effectively countering a mass Protoss air army as a Terran player is to use a combination of defensive structures, ground units, and air units to weaken the enemy army and protect the Terran player's base.

I mean... it's not *wrong*...

About Community

r/starcraft

All about the StarCraft games and professional scenes surrounding them. Please read the rules before submitting content.

Created Dec 10, 2008

291k Members • 324 Online

[Join](#)

meta_subliminal · 25 days ago

Well it's is wrong about the marauder part. It sounds like an elementary student being introduced to the 5 paragraph essay structure haha

6 Reply Share Report Save Follow

jiraiyaperv · 24 days ago

play like maru

1 Reply Share Report Save Follow

Comment removed by moderator · 21 days ago





First wave AI: Model-based

The Economist Menu Weekly edition Search Subscribe

Technology Quarterly Automobiles

Driverless cars show the limits of today's AI

Jun 13th 2020 edition >

They, and many other such systems, still struggle to handle the unexpected

A cartoon illustration by Tom Gauld shows a blue driverless car with a person inside looking at a tablet. Above the car are three thought bubbles containing question marks and exclamation points, indicating confusion or uncertainty. The car is approaching a road sign that is tilted and appears to be melting or distorted, suggesting that the AI system is struggling to interpret the visual information correctly.

Tom Gauld

Second wave AI: Data-driven

“Learn from lots of data”

FOOLING THE AI

Deep neural networks (DNNs) are brilliant at image recognition — but they can be easily hacked.

These stickers made an artificial-intelligence system read this stop sign as 'speed limit 45'.

©nature

1956

2011

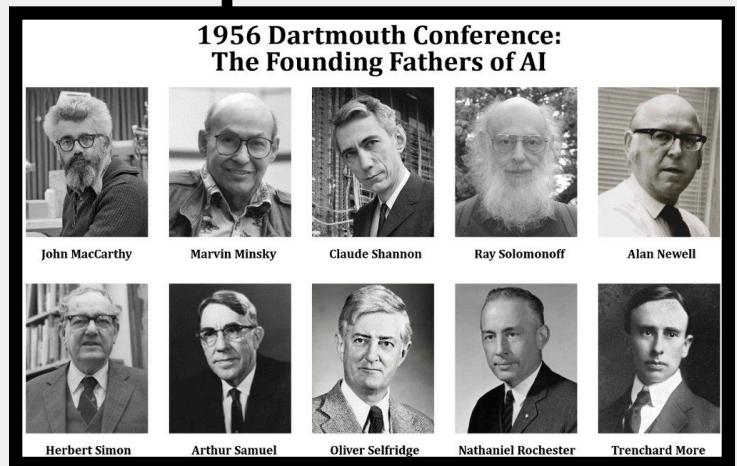
Time





First wave AI: Model-based

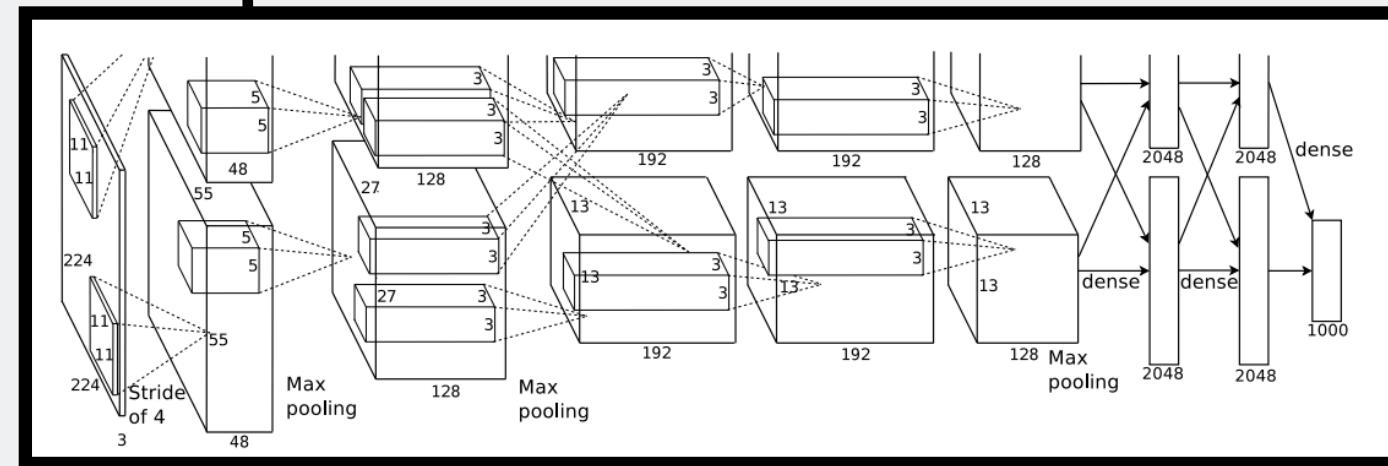
“Think through the entire problem”



1956

Second wave AI: Data-driven

“Learn from lots of data”



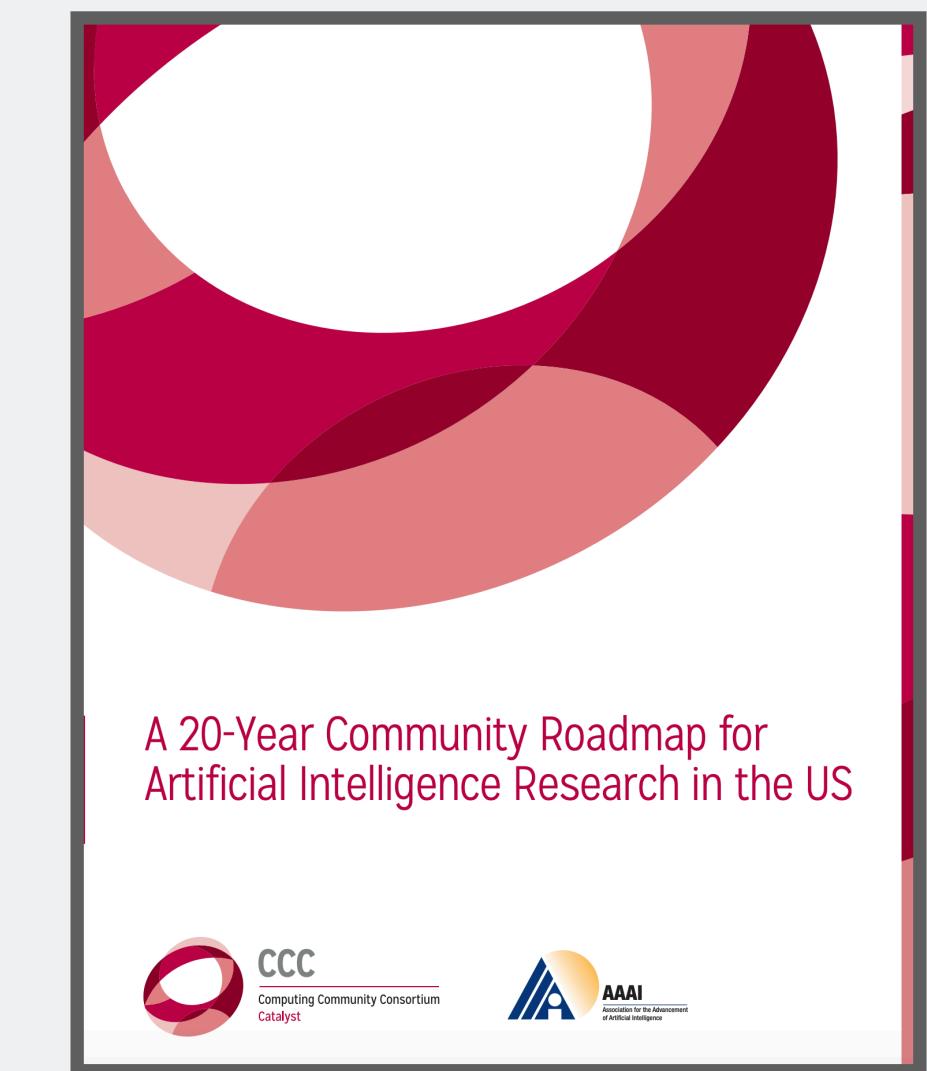
2011

Third wave AI: Explainable

“Combine first and second wave AI to generate explanations”



20??



Time

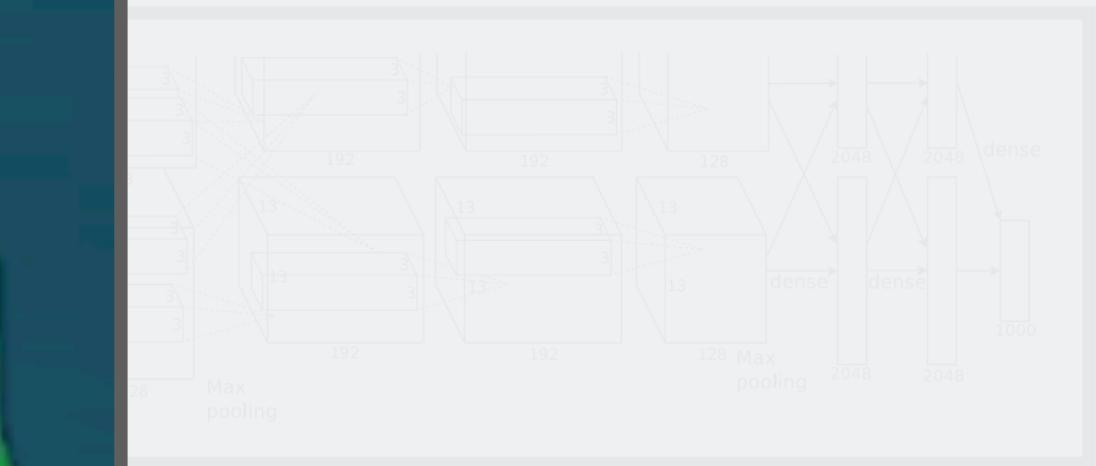




First wave AI: Model-based

"Think through the entire problem"

Is this handwritten character a '9' or a '4'?



Third wave AI: Explainable

"Combine first and second wave AI to generate explanations"



1956 2011 20?? Time



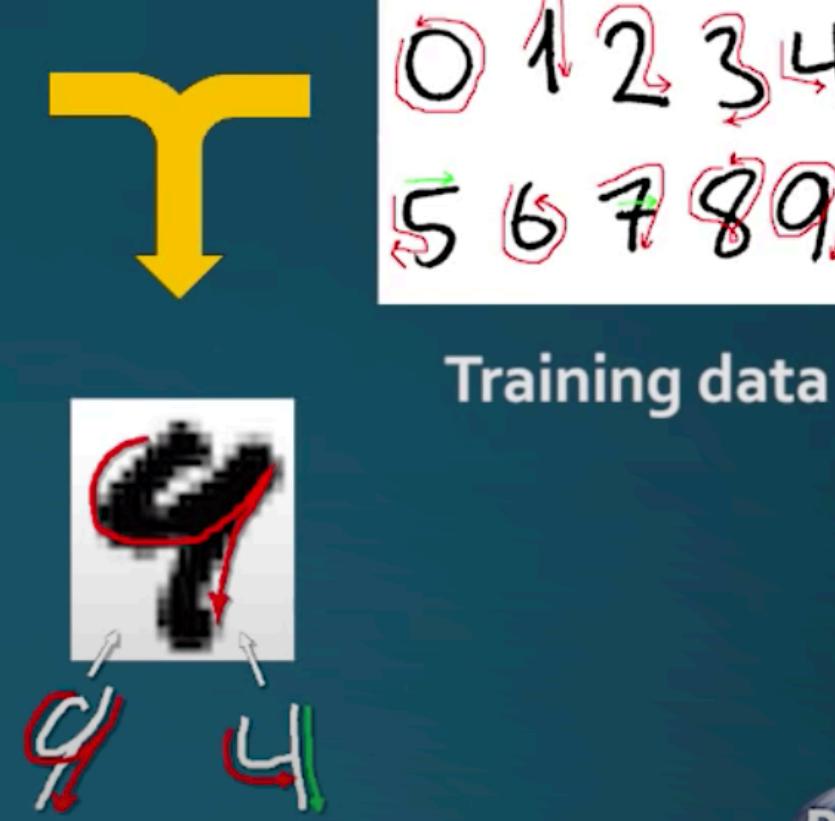


Models to drive decisions

Probable number of strokes: 1 - 4
Each stroke: probable trajectory
Each trajectory: probable shift in shape and location

Seed model

Generative model
Generates explanations of how a test character might have been created



▶ ▶ 🔍 14:54 / 16:11

▶ CC ⚙️ 🔍

A DARPA Perspective on Artificial Intelligence

241,472 views • Feb 15, 2017

5.5K 130 SHARE SAVE ...

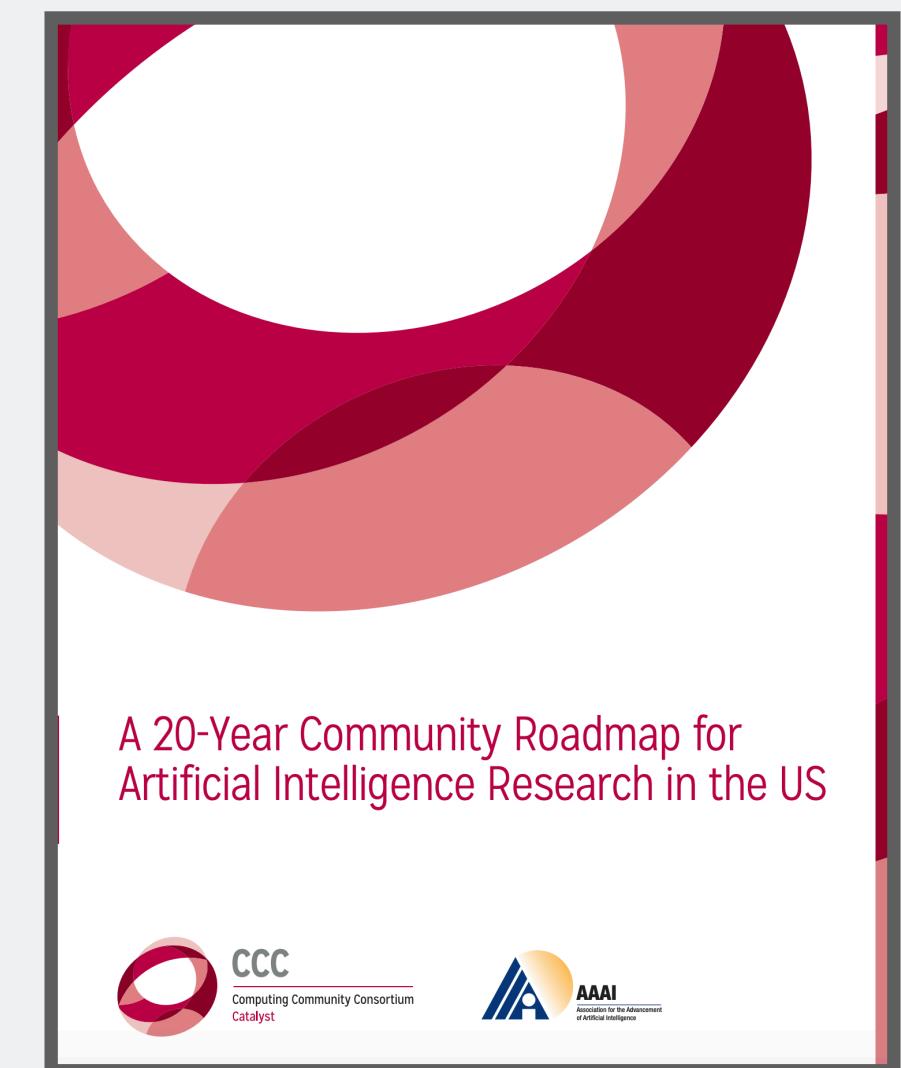


DARPAtv ✓
143K subscribers

SUBSCRIBE

Third wave AI: Explainable

"Combine first and second wave AI to generate explanations"



1956

2011

20??

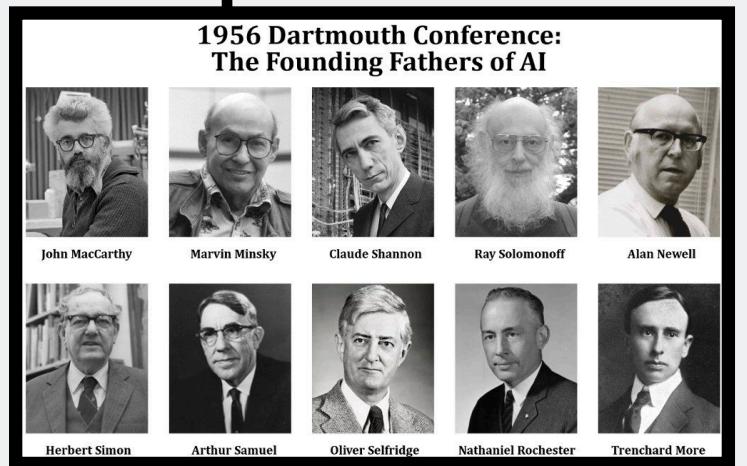
Time





First wave AI: Model-based

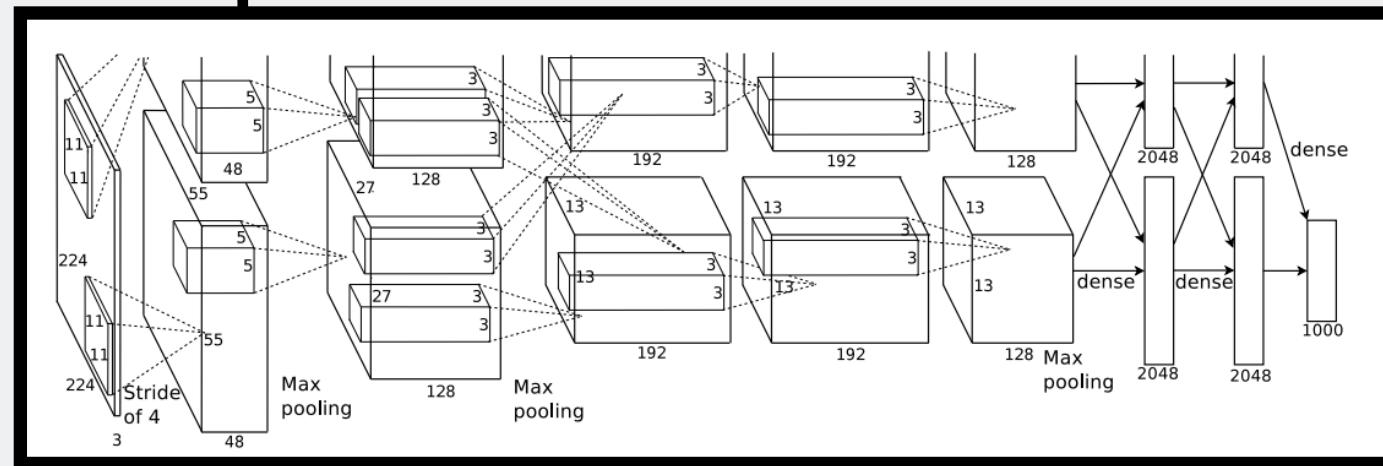
“Think through the entire problem”



1956

Second wave AI: Data-driven

“Learn from lots of data”



2011

Third wave AI: Explainable

“Combine first and second wave AI to generate explanations”



20??

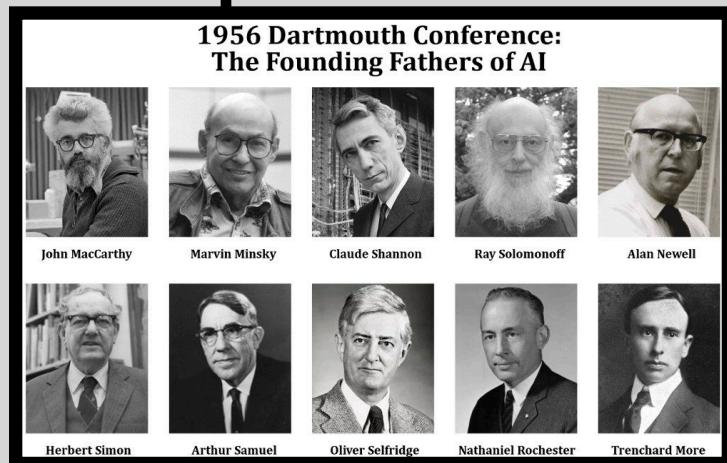
Time



Other Robotics and AI courses

First wave AI: Model-based

“Think through the entire problem”

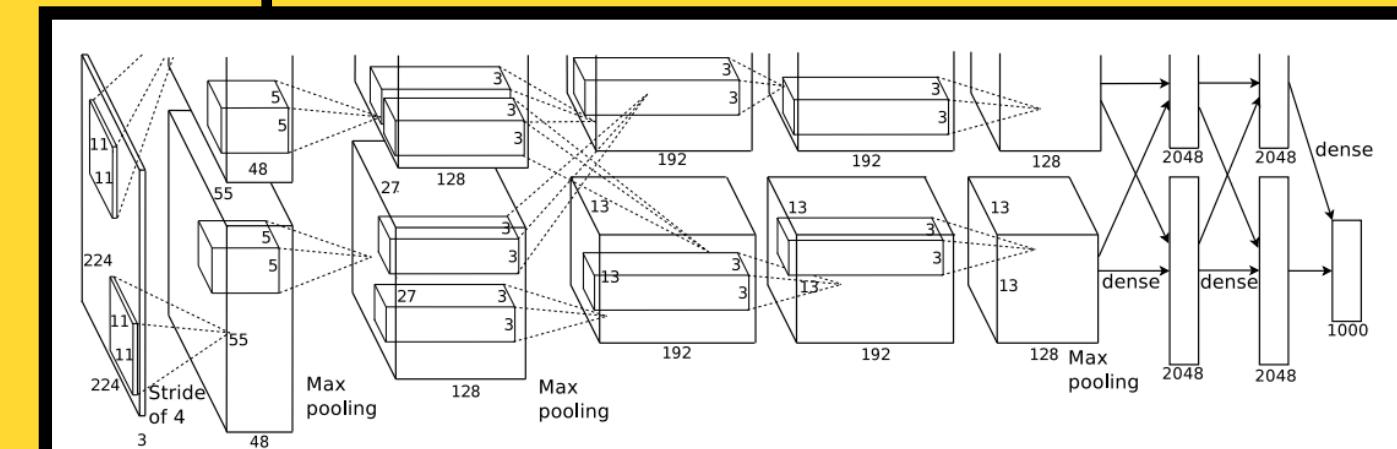


1956

DeepRob is a step into modern robot learning

Second wave AI: Data-driven

“Learn from lots of data”

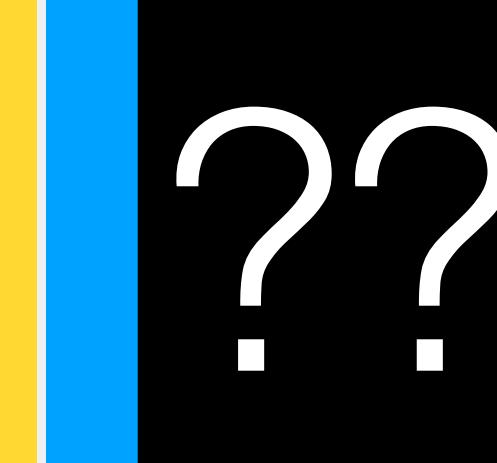
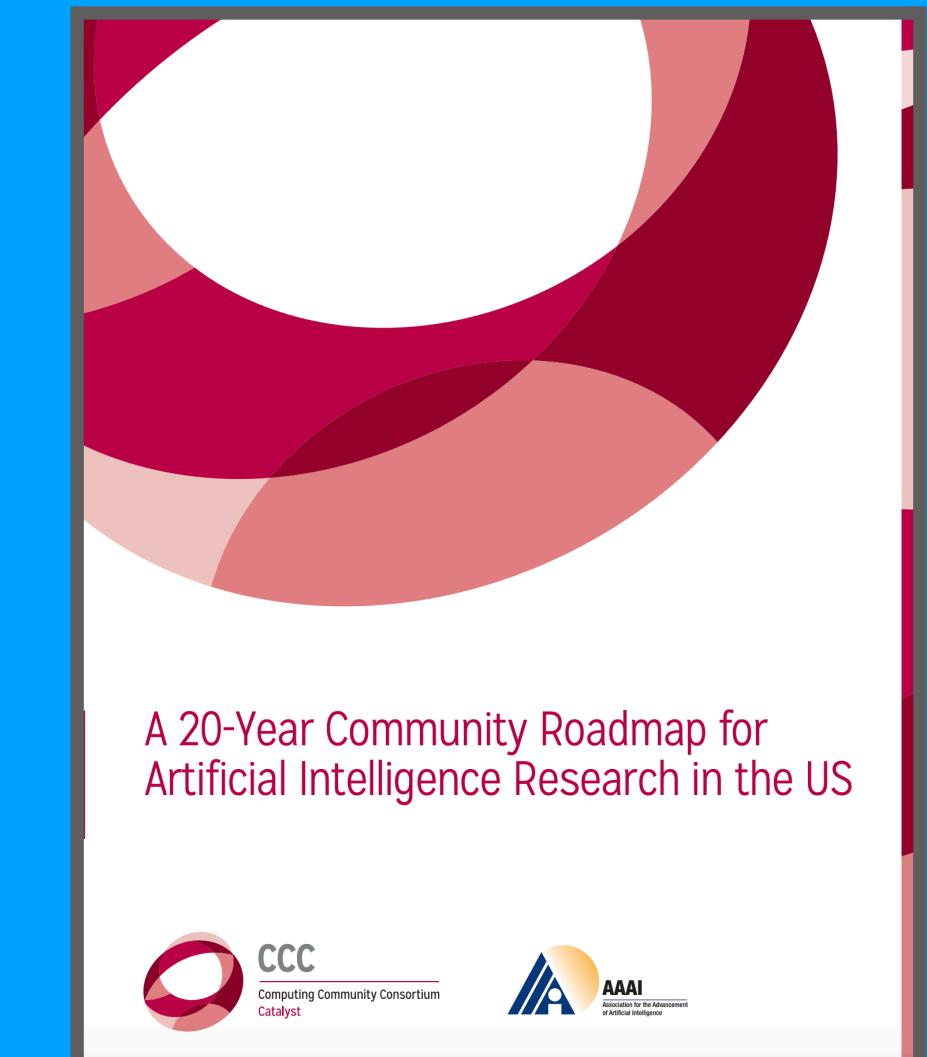


2011

Research for future AI

Third wave AI: Explainable

“Combine first and second wave AI to generate explanations”



20??

Time

DeepRob Course Staff



Anthony Opiari



Prof. Chad Jenkins



Prof. Karthik Desingh



Huijie Zhang
Research Associate



Jiyue Zhu
Instructional Aide

Course Resources

- Course Website: <https://deepprob.org>
 - Syllabus, calendar, project files, slides, links, etc.
- Ed Stem: <https://edstem.org/us/courses/31008/discussion/>
 - Forum for communication and question answering
 - Optional to join and shared across course offerings
- Autograder: <https://autograder.io/web/course/199>
 - Project submission and grading



Course Website—DeepRob.org

The screenshot shows a web browser window for the URL deeprob.org. The page title is "Home | Deep Rob". The main content area features a large heading "Deep Learning for Robot Perception" and subtext about the course covering neural-network-based deep learning for robot perception. A sidebar on the left lists navigation links: Home (which is active), Syllabus, Calendar, Projects, and Staff. A note at the bottom of the sidebar states: "This site uses Just the Docs, a documentation theme for Jekyll." The top right of the page includes a search bar, navigation links for Forum, Office Hours, Autograder, and Gradescope, and the University of Michigan Robotics logo.

DR Home | Deep Rob

deeprob.org

Deep Rob

Search Deep Rob

Forum Office Hours Autograder Gradescope

M ROBOTICS
UNIVERSITY OF MICHIGAN

Deep Learning for Robot Perception

ROB 498-002 & 599-009, Winter 2023 at The University of Michigan

This course covers the necessary background of neural-network-based deep learning for robot perception – building on advancements in computer vision that enable robots to physically manipulate objects. During the first part of this course, students will learn to implement, train and debug their own neural networks. During the second part of this course, students will explore recent emerging topics in deep learning for robot perception and manipulation. This exploration will include analysis of research publications in the area, building up to reproducing one of these publications for implementation as a final course project.

This site uses [Just the Docs](#), a documentation theme for Jekyll.

This course is being offered through a Distributed Teaching Collaborative between faculty at the University of Michigan ([Anthony Otipari](#), [Chad Jenkins](#)) and the University of Minnesota ([Karthik Desingh](#)).



Meeting Logistics

- In-person Lectures
 - Tue/Th 3:00-4:30 PM EST
 - [Zoom \(Meeting ID: 936 5222 5304\)](#)
 - Recordings available on course website ([deepprob.org](#))
- Discussion Section
 - Fri 4:30-5:30 PM EST
 - [Zoom \(Meeting ID: 936 5222 5304\)](#)
 - Tutorials and paper discussions

- Office Hours



Anthony

Wed, 1:30-3:00pm, **3320 FMCRB**
Fri, 9:00-10:30am, **3320 FMCRB**



Huijie

Mon, 2:00-3:00pm, **2320 FMCRB**
Tue, 2:00-3:00pm, **2320 FMCRB**

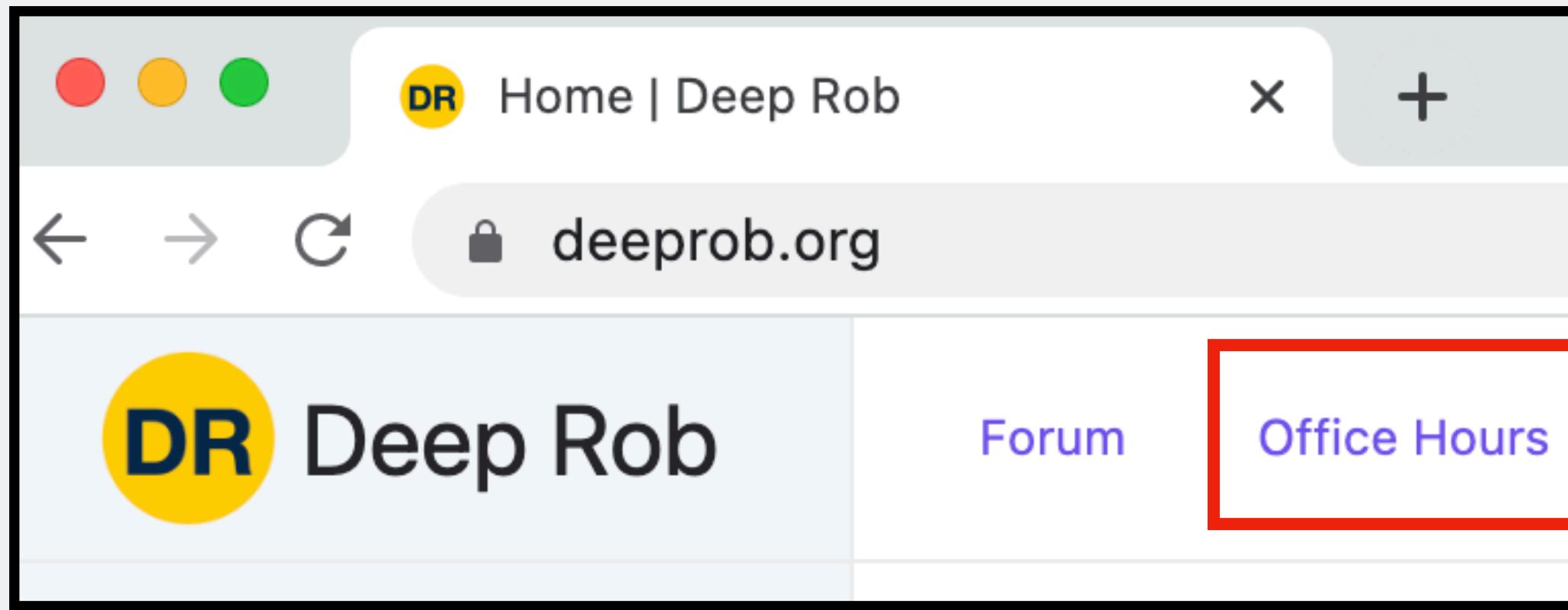


Jiyue

Th, 1:00-3:00pm, **2320 FMCRB**



Office Hours Queue



**Online queue management
tool to ensure efficient use
of the resource**

A screenshot of a web browser window titled "ROB 498-002/599-009 Office". The URL is "eecssoh.eecs.umich.edu/queues/2HK9nZeBgJBEPF6k84hbys3PNoK". The page title is "EECS Office Hours". It features a large blue box with a white heart icon and the text "The queue is empty. We're lonely over here!". To the right, there's a "Queue" section with a staff count of 0 and a note that the queue will be closed manually by staff. There's also a "Sign Up" section with fields for "Description" (with placeholder text "Help us help you—please be descriptive") and "Location", and a "Sign Up" button.



Course Structure

- Objective: Give you the computational skills to understand and reproduce emerging applications of deep learning for robot perception
- Project focused class
 - 6 total projects: building in complexity from basic linear layers to state-of-the-art research topics

Course Structure

- First half: Fundamentals
 - How to implement, train and debug neural networks
 - Classifiers, fully-connected networks, convolutional networks
 - Weeks 1-6 and Projects 0-2
- Second half: Emerging topics
 - Seminar style coverage of active research areas in robotics
 - 3D perception, pose estimation, tracking, NeRFs, transformers
 - Weeks 7-14 and Projects 3 onward



Schedule—DeepRob.org/calendar

The screenshot shows a web browser window for the 'Calendar | Deep Rob' page at deeprob.org/calendar/. The page features a navigation bar with links for Home, Syllabus, Calendar (which is highlighted), Projects, and Staff. The main content area is titled 'Calendar' and displays a weekly schedule:

- Week 1:**
 - Jan 5: LEC 1 Course Introduction
PROJECT 0 OUT
 - Jan 6: DIS 1 Intro to Python, Pytorch and Colab
- Week 2:**
 - Jan 10: LEC 2 Image Classification
 - Jan 12: LEC 3 Linear Classifiers
PROJECT 0 DUE PROJECT 1 OUT
 - Jan 13: DIS 2 Intro to PROPS Dataset
- Week 3:**
 - Jan 17: LEC 4 Regularization + Optimization
 - Jan 19: LEC 5 Neural Networks
 - Jan 20: DIS 3 Neural Networks in PyTorch

A note at the bottom left states: "This site uses Just the Docs, a documentation theme for Jekyll."

- Lecture slides
- Discussion resources
- Suggested readings
- Project schedule



Project Topics

- Project 0
 - Introduction to Python and PyTorch
- Project 1
 - Classification using K-Nearest Neighbors and Linear Models
- Project 2
 - Classification using Fully-Connected and Convolutional Neural Networks
- Project 3
 - Detection using convolutional neural networks
- Project 4
 - State estimation with deep networks
- Final Project
 - Published paper presentation, reproduction, extension and report

Final Project

- Published paper presentation, reproduction, extension and report
- Work as collaborative teams up to 3 people
- Deliverables:
 - Paper presentation during lecture
 - Result reproduction as code
 - Network extension as code
 - Project report as <= 2-page paper



Project Grading

- Projects 0-4 graded with [autograder.io](#)
 - 2 total late days available
 - 25% daily penalty after deadline and late days
- Final project graded manually by course staff

Overall Grading Policy

- Projects 0-4: 12% each
- Final Project: 24%
 - Presentation: 6%
 - Reproduction: 6%
 - Extension: 6%
 - Written report: 6%
- 16 Pre-Lecture Quizzes: 1% each

Collaboration Policy

- All work submitted must be your own
 - All code submitted must comply with College of Engineering Honor Code
- No code can be communicated, including verbally
 - Explicit use of external sources must be clearly cited
- Free flow of discussion and ideas is encouraged



Discussion Forum

- Ed Stem available for course discussion and questions
 - Forum is shared across UMich and UMinn students
 - Participation and use is not required
 - Opt-in using [this Google form](#)
 - **Discussion of quizzes and verbatim code must be private**



Project 0

- Instructions and code available on the website
- Released today: deeprob.org/projects/project0/
- Intro at this week's discussion
- Due next Thursday, January 12th 11:59 PM EST





Project 0

The screenshot shows a Google Colab notebook titled "pytorch101.ipynb". The left sidebar contains a "Table of contents" with sections like "ROB 498-002/599-009 Project 0-1: PyTorch 101", "Setup Code", "Google Colab Setup", "Introduction", "Python 3", "PyTorch", "Tensor Basics", "Datatypes", "Tensor indexing", "Slice indexing", "Integer tensor indexing", "Boolean tensor indexing", "Reshaping operations", and "View". The main content area displays the "Setup Code" section, which includes instructions for putting name and UMID, a "Your Answer:" field, and boilerplate code for autoreloading Python files.

ROB 498-002/599-009 Project 0-1: PyTorch 101

Before we start, please put your name and UMID in following format
: Firstname LASTNAME, #00000000 // e.g.) Anthony OPIPARI, #12345678

Your Answer:
Your NAME, #XXXXXXXX

Setup Code

Before getting started we need to run some boilerplate code to set up our environment. You'll need to rerun this setup code each time you start the notebook.

First, run this cell load the [autoreload](#) extension. This allows us to edit .py source files, and re-import them into the notebook for a seamless editing and debugging experience.

```
[ ] %load_ext autoreload
%autoreload 2
```

The autoreload extension is already loaded. To reload it, use:
%reload_ext autoreload

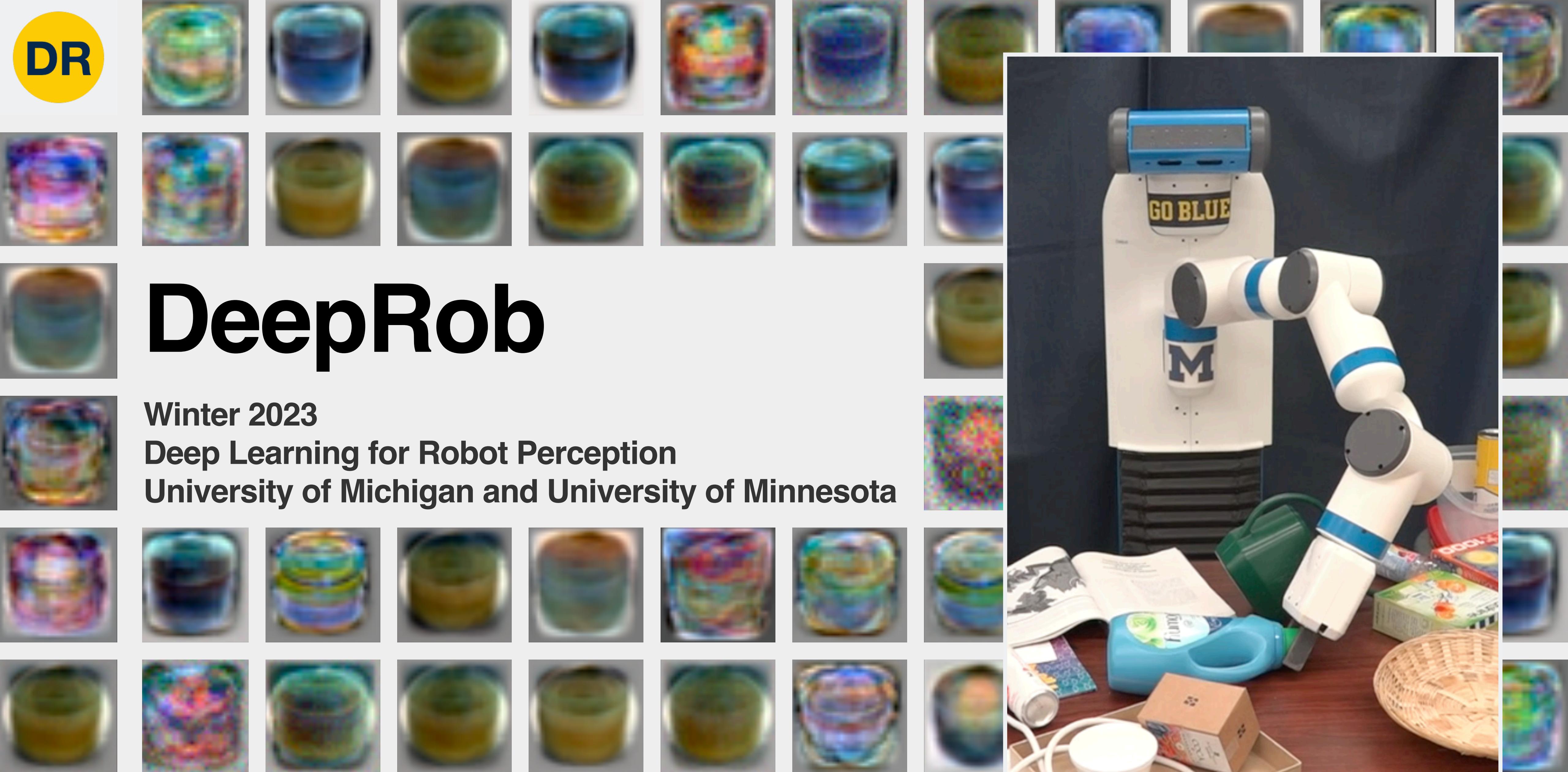


Next Week: Classification

Image Classification and Linear Classifiers



DR



DeepRob

Winter 2023

Deep Learning for Robot Perception

University of Michigan and University of Minnesota